User's Manual

IMO GM7 Series

Contents

Chapter 1. General.....1-1~1-6

1.1 Guide to Use this Manual 1-1

1.2 Features.....1-2

1.3 Terminology 1-4

Chapter 2. System Configuration.....2-1~2-6

2.1 Overall Configuration.....2-1

2.1.1 Basic system 2-1

2.1.2 Cnet I/F System 2-2

2.2 Product functional model 2-4

2.2.1 Product function Block.....2-4

2.2.2 GM7 Series System Equipment Product.....2-5

Chapter 3. General Specification.....3-1

3.1 General specifications 3-1

Chapter 4. Names of Parts .....4-1~4-6

4.1 Base Unit 4-1

4.1.1 10-point base unit.....4-3

4.1.2 20-point base unit.....4-3

4.1.3 30-points Base Unit.....4-3

4.1.4 40-Points Base Unit.....4-4

4.1.5 60-Points Base Unit.....4-4

4.2 Expansion Module.....4-5

4.2.1 Digital I/O Module 4-5

4.2.2 A/D·D/A Combination Module 4-5

4.2.3 Analogue timer Module 4-5

4.2.4 Option Module 4-6

Chapter 5. CPU .....5-1~5-48

5.1 Power Supply Specifications 5-1

5.2 CPU Specifications.....5-2

5.3 Operation Processing 5-4

5.3.1 Operation Processing Method 5-4
5.3.2 Operation Processing at momentary power failure occurrence 5-5
5.3.3 Scan time 5-6
5.3.4 Scan-watchdog timer 5-6
5.3.5 Timer processing 5-7
5.3.6 Counter processing 5-9

5.4 Program 5-11

5.4.1 Program configuration.....5-11
5.4.2 Program execution procedure 5-12
5.4.3 Task.....5-15
5.4.4 Error handling.....5–22
5.4.5 Precautions when using special modules.....5–23

5.5 Operation modes 5-24

5.5.1 RUN mode 5-24
5.5.2 STOP mode 5-25
5.5.3 PAUSE mode.....5-25
5.5.4 DEBUG mode 5-25
5.5.5 Operation mode Change.....5-26

5.6 Functions 5-28

5.6.1 Restart mode.....5-28
5.6.2 Self-diagnosis.....5-30
5.6.3 Remote function 5-30
5.6.4 I/O Force On/Off function 5-31
5.6.5 Direct I/O operation function.....5-32
5.6.6 External device error diagnosis function 5-32

5.7 Memory Configuration 5–35

5.8 I/O No. Allocation Method.....5–37

5.9 Built-in Flash Memory 5–37

5.9.1 Structure.....5-37
5.9.2 Usage.....5-38

5.10 External Memory Module.....5–40

5.10.1 Structure.....5-40
5.10.2 Usage.....5-40

5.11 Battery 5–43

5.12 RTC module 5–44

Chapter 6. Input and Output Modules.....6-1~6-13

6.1 Input and Output Specifications 6-1

6.2 Digital Input Specifications 6–2

6.2.1 Base Unit.....6-2

6.2.2 Extended Module.....6-6

6.3 Digital output Specifications 6–7

6.3.1 Base unit (Relay output).....6-7

6.3.2 Base unit (Transistor output).....6-10

6.3.2 Extended Module.....6-13

Chapter 7. Usage of Various Functions.....7-1~7-52

7.1 Built-in function 7–1

7.1.1 High-speed counter function 7-1

7.1.2 Pulse Output Function.....7-8

7.1.3 Pulse Catch function.....7-18

7.1.4 Input Filter function.....7-20

7.1.5 PID Control function 7-21

7.1.6 External Interrupt function 7-40

7.2 Special Module 7–42

7.2.1 A/D-D/A Combination 7–42

7.2. 2 Analogue Timer 7–50

Chapter 8. Communication Function.....8-1~8-115

8.1 Direct Protocol Communication 8–1

8.1.1 Introduction 8-1

8.1.2 System Configuration method.....8-2

8.1.3 Frame Structure.....8-5

8.1.4 List of Commands.....8-8

8.1.5 Data Type.....8-9

8.1.6 Execution of Commands 8-10

8.1.7 1:1 Built-in Communication between GM7's.....8-30

8.1.8 Error Codes.....8-48

8.2 User Defined Protocol Communication 8–50

8.2.1 Introduction 8-50

8.2.2 Parameter Setting....8-51
8.2.3 Function Block 8-58
8.2.4 Example of Use 1) 8-59
8.2.5 Example of Use 2) 8-76

8.3 Modbus Protocol Communication....8–85
8.3.1 Introduction 8-85
8.3.2 Basic Size....8-85
8.3.3 Parameter Setting....8-89
8.3.4 Function Block 8-91
8.3.5 Example of Use....8-108

Chapter 9. Installation and Wiring....9-1~9-11

9.1 Installation....9-1
9.1.1 Installation Environment 9-1
9.1.2 Handling Instructions 9-4
9.1.3 Connection of expansion module....9-7

9.2 Wiring....9-8
9.2.1 Power supply Wiring....9-8
9.2.2 I/O devices Wiring 9-10
9.2.3 Grounding....9-10
9.2.4 Cable Specifications for Wiring 9-11

Chapter 10. Maintenance ....10-1~10-2

10.1 Maintenance and Inspection 10-1

10.2 Daily Inspection....10-1

10.3 Periodic Inspection 10-2

Chapter 11. Trouble Shooting ..... 11-1~11-13

11.1 Basic Procedures of Troubleshooting 11-1

11.2 Troubleshooting 11-1

11.2.1 Troubleshooting flowchart used when the power LED turns off . 11-2

11.2.2 Troubleshooting flowchart used when the error LED is flickering 11-3

11.2.3 Troubleshooting flowchart used when the RUN LED turns off 11-4

11.2.4 Troubleshooting flowchart used when the I/O devices doesn't operate normally 11-5

11.2.5 Troubleshooting flowchart used when a program can't be written to the CPU 11-7

11.3 Troubleshooting Questionnaire 11-8

11.4 Troubleshooting Examples 11-9

11.4.1 Input circuit troubles and corrective actions 11-9

11.4.2 Output circuit troubles and corrective actions 11-10

11.5 Error code list 11-12

Appendix ..... App1-1~App4-1

Appendix 1 System definitions App1-1

Appendix 2 Flag list App2-1

Appendix 3 Function / Function block list App3-1

Appendix 4 Dimensions App4-1

Chapter 1. General

1.1 How to Use This Manual

This manual includes specifications, functions and handling instructions for the -GM7 series PLC.
This manual is divided up into chapters as follows:

Chapters	Title	Contents
Chapter 1	General	Describes configuration of this manual, unit's features and terminology.
Chapter 2	System configuration	Describes available units and system configurations in the -GM7 series.
Chapter 3	General Specification	Describes general specifications of units used in the -GM7 series.
Chapter 4	Names of Parts	Describes each kind of manufacturing goods, titles, and main functions
Chapter 5	CPU Part	Describes each kind of manufactured goods' usage
Chapter 6	Digital Input and Output Parts	
Chapter 7	Guides on Each Function	
Chapter 8	Communications Function	Describes built-in communication functions
Chapter 9	Installation and Writing	Describes installation, wiring and handling instructions for reliability of the PLC system
Chapter 10	Maintenance and Inspection	Describes the check items and method for long-term normal operation of the PLC system.
Chapter 11	Troubleshooting	Describes various operation errors and corrective actions.
Appendix1	System Definition	Describes parameter setting for basic I/O and communications module
Appendix 2	Flag List	Describes the types and contents of various flags.
Appendix 3	Function / Function Block List	Describes the types and processing time of function/function block.
Appendix 4	Dimensions	Shows dimensions of the base units and expansion modules

REMARK

1) This manual does not describe the programming method. For their own functions, refer to the related user's manuals.

1.2. Features

1) -GM series features

(1) Design on the basis of international standard specifications (IEC61131-3)

- Easy programming device support
- Language in compliance with IEC61131-3 are given (IL / LD / SFC)

(2) Open network by us of communications protocol in compliance with international standard specifications.

(3) High speed processing with an operation-dedicated processor included.

(4) Various special modules that enlarge the range of application of the PLC

2) GM7 series is extremely compact, to fit a wide range of applications.

(1) High speed processing

High speed processing of 0.5μs/step with an operation-dedicated processor included.

(2) Various built-in functions

The base unit can perform many functions without using separate modules.

It is possible to construct various systems just using the base unit.

- Fast Processing Applications

-Pulse catch: Allows the base unit to read 4 inputs, each having a pulse width as small as 0.2ms

-High speed counter: Support high-speed counting up to 1 phase 16kHz, 2 phase 8kHz.

-External interrupts : Using in applications that have a high-priority event which requires immediate responses.

- The input filter function help reduce the possibility of false input conditions from external noise, such as signal chattering. The filter time can be programmed from 0 to 15 ms.
- Using built-in pulse output without separate positioning module, it can control stepping motor or servo motor.
- Using RS-232C built-in port, it can connect with external devices, such as computers or monitoring devices and communicate 1:1 with GM7 or GM6 system.
- Using RS-485 built-in port, it can connect with external devices, such as computers or monitoring devices and communicate 1:N with GM7 or GM6 system. (10-point base unit only)
- It has PID control function with which it can easily constitute a system without separate module.

(3) It can easily do On/Off of the system, using RUN/STOP switch.

(4) It can constitute various system, using separate Cnet I/F module. (10-points main unit can not

(5) It can easily save the user program by simple manipulation in GMWIN.

(6) Strong self-diagnostic functions

It can detect the cause of errors with more detailed error codes.

(7) It can prevent unintentional reading and writing, using password.

(8) Restart mode setting function

It has cold and warm mode that it can be set for the convenience of the users.

(9) Debugging function

On-line debugging is available if the PLC Operation mode is set to debug mode.

- executed by one command.
- executed by break-point settings.
- executed by the condition of the device
- executed by the specified scan time.

(10) Various program execution function

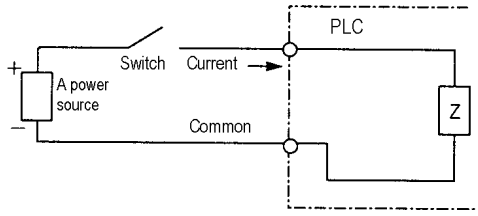
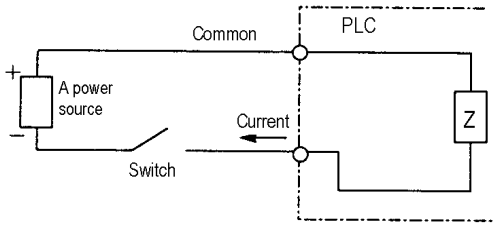
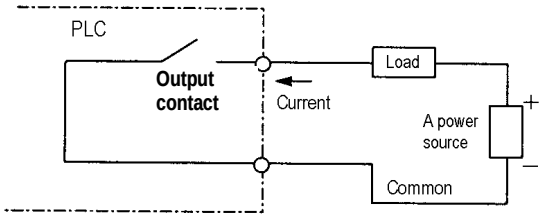
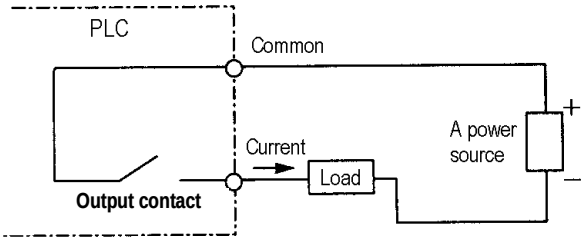
Time driven task, external and internal task program as well as scan program can be executed by setting the execution condition.

The user can set variously the program execution mode.

1.3 Terminology

The following table gives definition of terms used in this manual.

Terms	Definition	Remarks
Module	A standard element that has a specified function which configures the system. Devices such as I/O board, which inserted onto the mother board or base unit.	Example) CPU module Power Supply module I/O module
PLC system	A system which consists of the PLC and peripheral devices. A user program can control the system.	
Cold Restart	To restart the PLC system and user programs after all of the data(Variables and programs of I/O image area, of internal register, of timer of counter) were set to the specified conditions automatically or manually.	
Warm Restart	In the warm restart mode, The power supply Off occurrence will be informed to the user program and the PLC system restarts with the previous user-defined data and user program after the power supply Off.	
I/O Image Area	Internal memory area of the CPU module which used to hold I/O statuses.	
Watch Dog Timer	Supervisors the pre-set execution times of programs and warns if a program is not completed within the pre-set time.	
Function	Operation Unit which outputs immediately its operation result of an input, while four arithmetic operations comparison operation store their results in the inside of instructions.	
Function Block	Operation Units which store operation result in the inside of instruction such as timer and counter and use the operation results which have been stored through many scans.	
Symbolic Variable	Variables used after the user's definition of their names and types. Declarations as 'INPUT_0' = %IX0.0.2, 'RESULT = %MD1234' makes INPUT_0 and RESULT be able to used instead of %IX0.0.2 and %MD123 in programming.	
GMWIN	A peripheral device for the -GM series. It executes program creation, edit, compile and debugging.	
FAM	Abbreviation of the word 'Factory Automation Monitoring S/W'. It is used to call S/W packages for process supervision.	
Task	It means startup conditions for a program. There are three types of periodic task, internal contact task and external contact task which starts by the input signals of external input modules.	

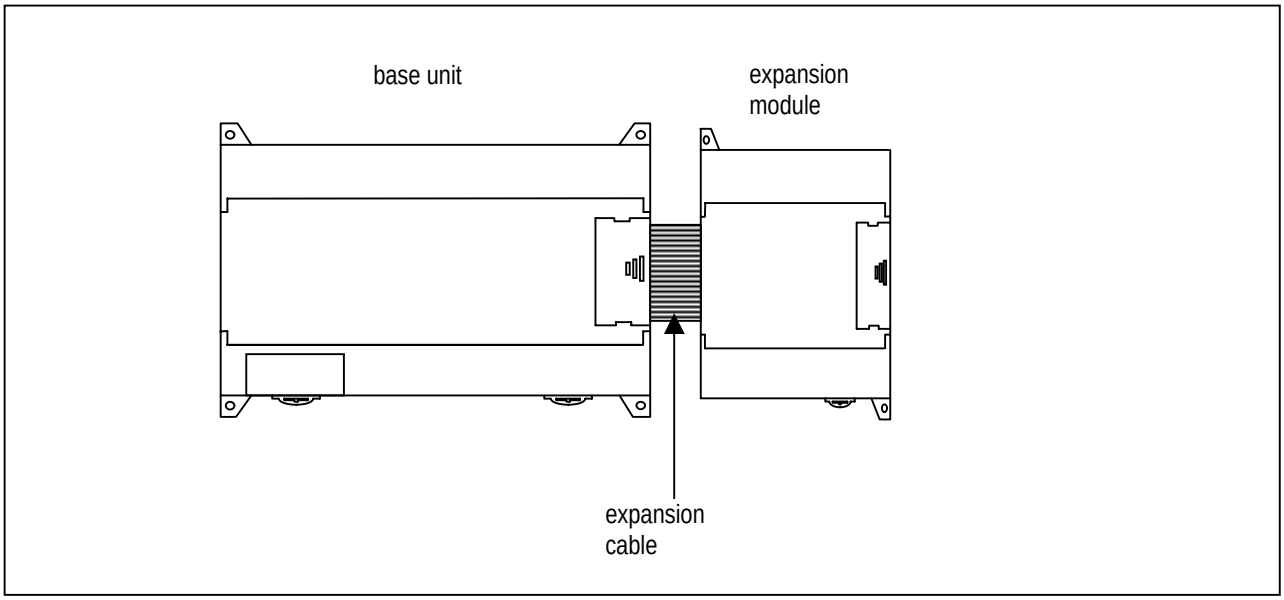
Terms	Definition	Remarks
Sink Input	Current flows from the switch to the PLC input terminal if a input signal turns on. 	
Source Input	Current flows from the PLC input terminal to the switch after a input signal turns on. 	
Sink Output	Current flows from the load to the output terminal and the PLC output turns on. 	
Source Output	Current flows from the output terminal to the load and the PLC output turns on. 	
Fnet	Fieldbus Network	
Cnet	Computer Network	
Dnet	DeviceNet Network	

Chapter 2. System Configuration

The GM7 series has suitable to configuration of the basic, computer link and network systems.
This chapter describes the configuration and features of each system.

2.1. Overall Configuration

2.1.1 Basic system



Total I/O points		• 10-80 points
Maximum numbers of expansion modules	Digital I/O module	• 2 modules
	A/D-D/A Composite module	• 2 modules
	Analog timer	• 3 modules
	Cnet I/F module	• 1 module [Not available for 10-point module]
Items	Base unit	•G7M-DR10A, G7M-DR20A, GM7-DR30A, G7M-DR40A, GM7-DR60A G7M-DR10A/DC, G7M-DR20A/DC, G7M-DR30A/DC, G7M-DR40A/DC, G7M-DR60A/DC, G7M-DT10A, G7M-DT20A, G7M-DT30A, G7M-DT40A, G7M-DT60A
	Expansion module: Digital I/O module	• G7E-DR10A
	Expansion module: A/D-D/A Composite module	• G7F-ADHA
	Expansion module: Analog timer	• G7F-AT2A
	Expansion module: Cnet I/F modules	• G7L-CUEB, G7L-CUEC

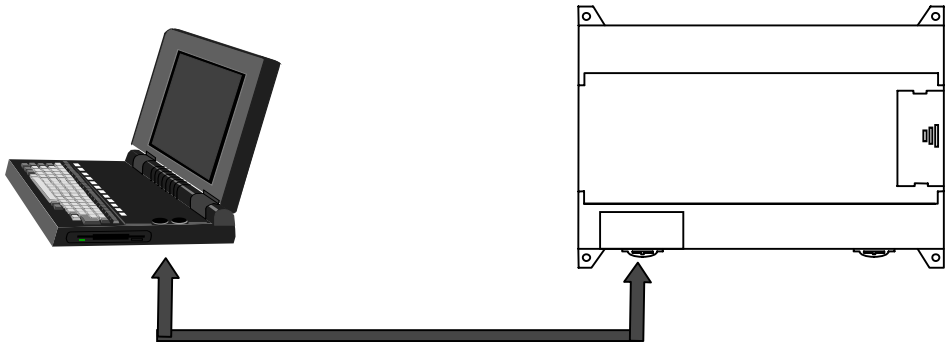
Total 3 modules

2.1.2 Cnet I/F system

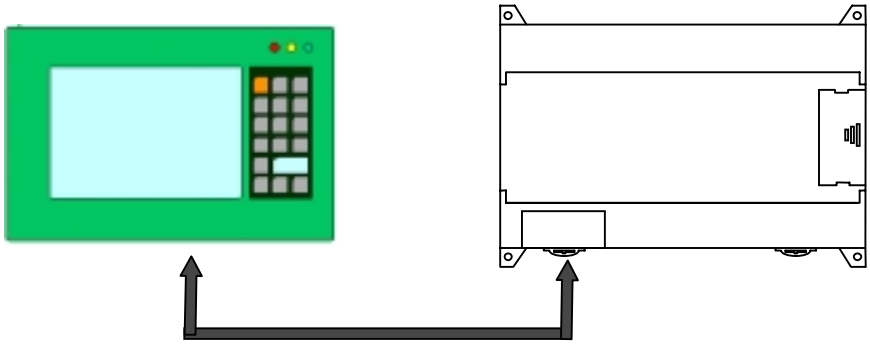
Cnet I/F System is used for communication between the base unit and external devices using RS-232C/RS-422 Interface. The GM7 has a built-in RS-232C port and has also G7L-CUEB for RS-232C, G7L-CUEC for RS-422. It is possible to construct communications systems on demand.

1) 1:1 Communications system

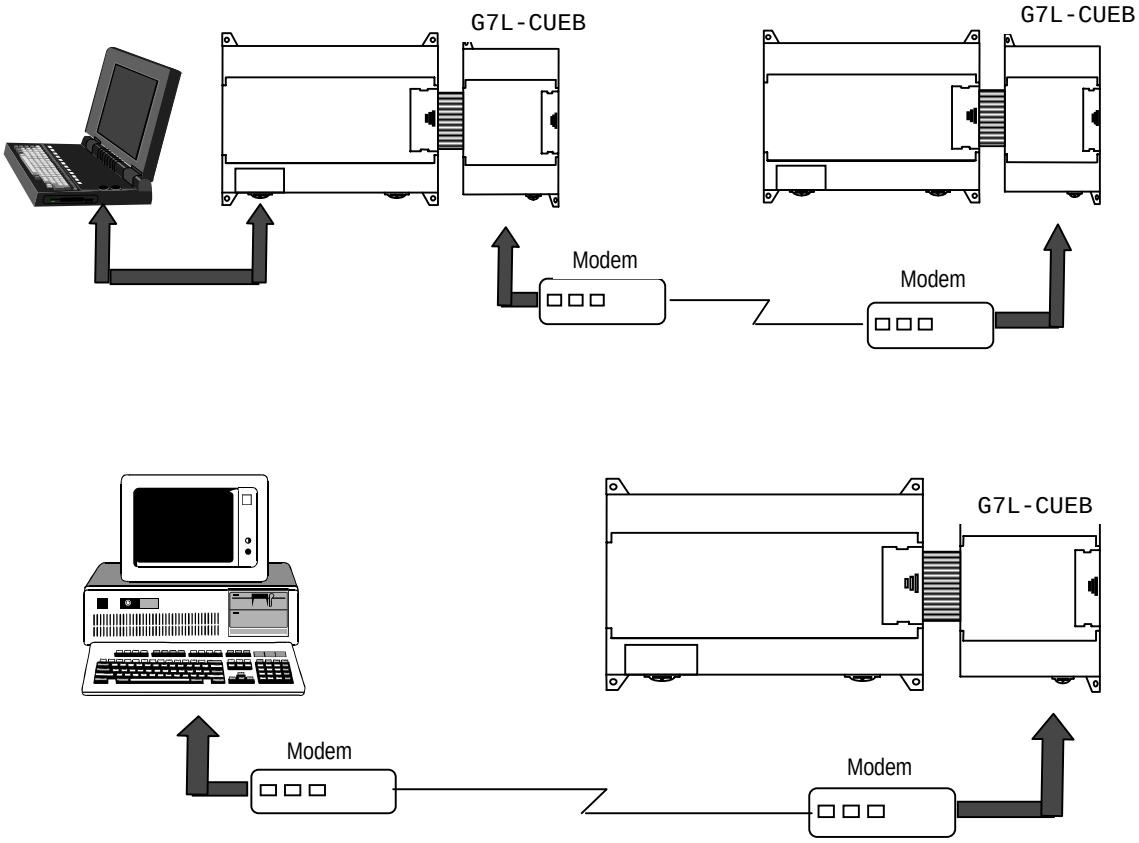
(1) 1:1 ratio of an external device (computer) to base unit using a built-in port



(2) 1:1 ratio of an external device (monitoring unit) to base unit using a built-in port



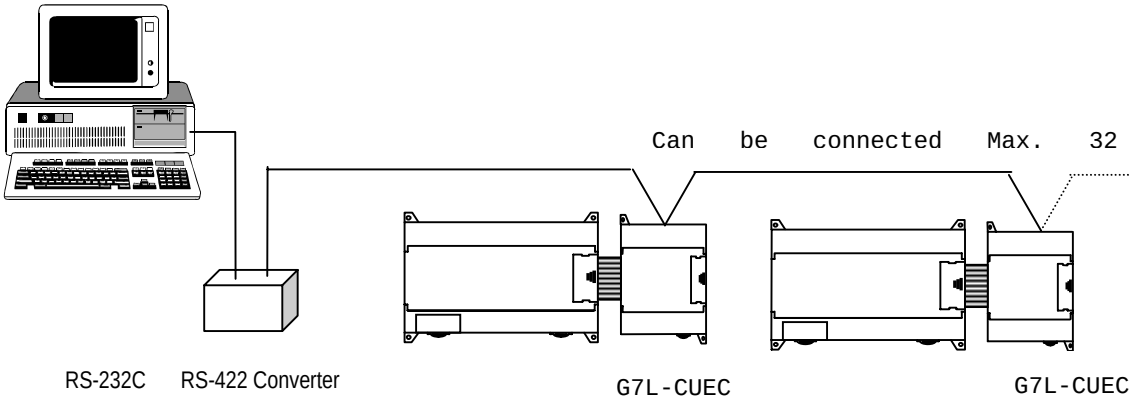
(3) RS-232C Communication over a long distance via modem by Cnet I/F modules



2) 1:n Communications system

This method can connect between one computer and multiple base units for up to 32 stations

(G7L-CUEA module is not required for 10-point base units)

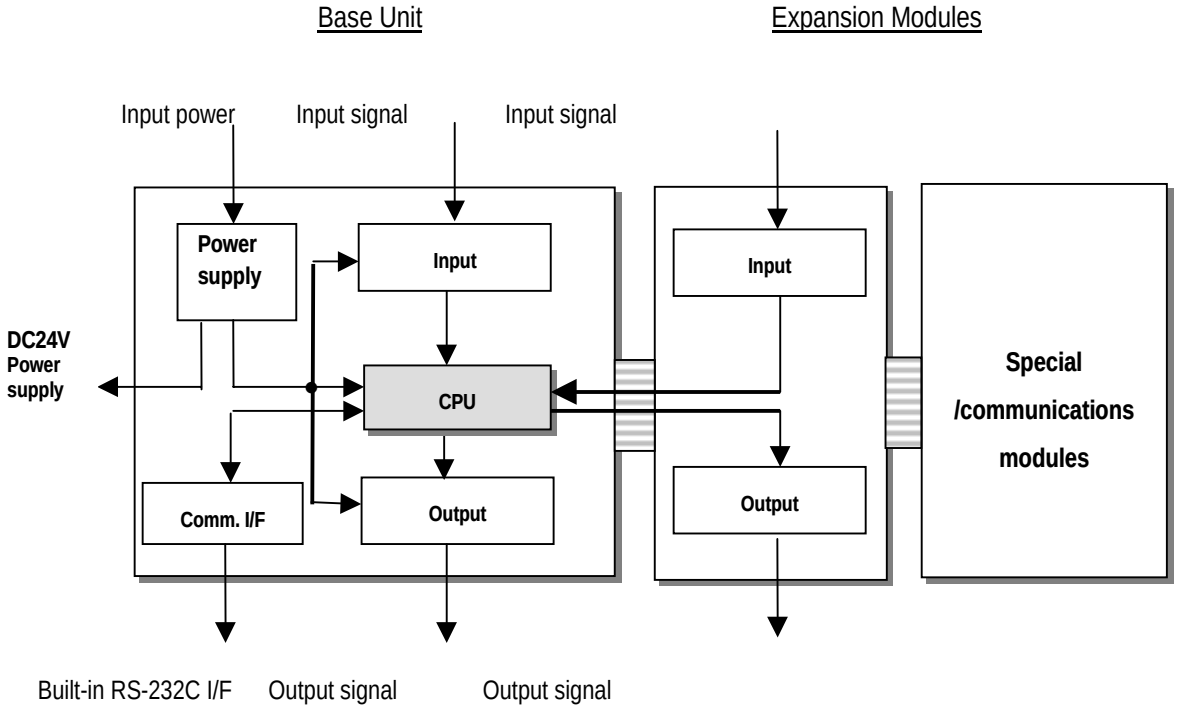


2.2 Product functional model

The following describes functional model of the -GM7 series.

2.2.1 Product Function Block

Product function block for the GM7 series is as follows.



Sub-system	Description
CPU	• Signal processing function •Operating system function •Application program storage / memory function •Data storage / memory function •Application program execution function
Input	• The input signals and/or data obtained from the machine/process to appropriate signal levels for processing
Output	• The output signals and/or data obtained from the signal processing function to appropriate signal levels to drive actuators and/or displays
Power Supply	• Provides for conversion and isolation of the PLC system power from the power supply
Communications Interface	• Provides the data exchange with other systems, such as GMWIN, computers

2.2.2 GM7 Series System Equipment

Section	Items	Models	Description	Remark
Basic	Base Unit	G7M-DR10A G7M-DR10A/DC G7M-DT10A	• I/O Points - 6 DC inputs / 4 relay outputs (G7M-DR10A, G7M-DR10A/DC) - 6 DC inputs / 4 transistor outputs (G7M-DT10A) • Program capacity : 68k bytes • Built-in function -High-speed counter : Phase1 16 kHz, phase2 8 kHz 1channel -pulse output : 1 × 2 kHz -pulse catch : pulse width 0.2ms, 4 points -external contact point interrupt: 0.4ms, 8points -input filter: 0 ~ 15ms (all input) -PID control function -RS-232C communication	
		G7M-DR20A G7M-DR20A/DC G7M-DT20A	• I/O Points - 12 DC inputs / 8 relay outputs (G7M-DR20A, G7M-DR20A/DC) - 12 DC inputs / 8 transistor outputs (G7M-DT20A) • Program capacity : 68k bytes • Built-in function -High-speed counter : Phase1 16 kHz, phase2 8 kHz 1channel -pulse output : 1 × 2 kHz -pulse catch : pulse width 0.2ms, 4 points -external contact point interrupt: 0.4ms, 8points -input filter: 0 ~ 15ms (all input) -PID control function -RS-232C communication	
		G7M-DR30A G7M-DR30A/DC G7M-DT30A	• I/O Points - 18 DC inputs / 12 relay outputs (G7M-DR30A, G7M-DR30A/DC) - 18 DC inputs / 12 transistor outputs (G7M-DT30A) • Program capacity : 68k bytes • Built-in function -High-speed counter : Phase1 16 kHz, phase2 8 kHz 1channel -pulse output : 1 × 2 kHz -pulse catch : pulse width 0.2ms, 4 points -external contact point interrupt: 0.4ms, 8points -input filter: 0 ~ 15ms (all input) -PID control function -RS-232C communication	

Section	Items	Models	Description	Remark
		G7M-DR40A G7M-DR40A/DC G7M-DT40A	• I/O Points - 24 DC inputs / 16 relay outputs (G7M-DR40A, G7M-DR40A/DC) - 24 DC inputs / 16 transistor outputs (G7M-DT40A) • Program capacity : 68k bytes • Built-in function -High-speed counter : Phase1 16 kHz, phase2 8 kHz 1channel -pulse output : 1 × 2 kHz -pulse catch : pulse width 0.2ms, 4 points -external contact point interrupt: 0.4ms, 8points -input filter: 0 ~ 15ms (all input) -PID control function -RS-232C communication	
		G7M-DR60A G7M-DR60A/DC G7M-DT60A	• I/O Points - 36 DC inputs / 24 relay outputs (G7M-DR60A, G7M-DR60A/DC) - 36 DC inputs / 24 transistor outputs (G7M-DT60A) • Program capacity : 68k bytes • Built-in function -High-speed counter : Phase1 16 kHz, phase2 8 kHz 1channel -pulse output : 1 × 2 kHz -pulse catch : pulse width 0.2ms, 4 points -external contact point interrupt: 0.4ms, 8points -input filter: 0 ~ 15ms (all input) -PID control function -RS-232C communication	
Expansion module	Digital I/O module	G7E-DR10A	• I/O points -6 DC inputs / 4 relay outputs	
	A/D · D/A Composite module	G7F-ADHA	• A/D : 2 channel , D/A : 1 channel	
	Analog timer module	G7F-AT2A	• Points : 4points • Digital output range : 0~200	
	Cnet I/F module	G7L-CUEB	• RS-232C : 1 channel	Not available for 10-poiunt units
		G7L-CUEC	• RS-422 : 1 channel	

Chapter 3. General Specifications

3.1 General specifications

The following shows the general specifications of the GM series.

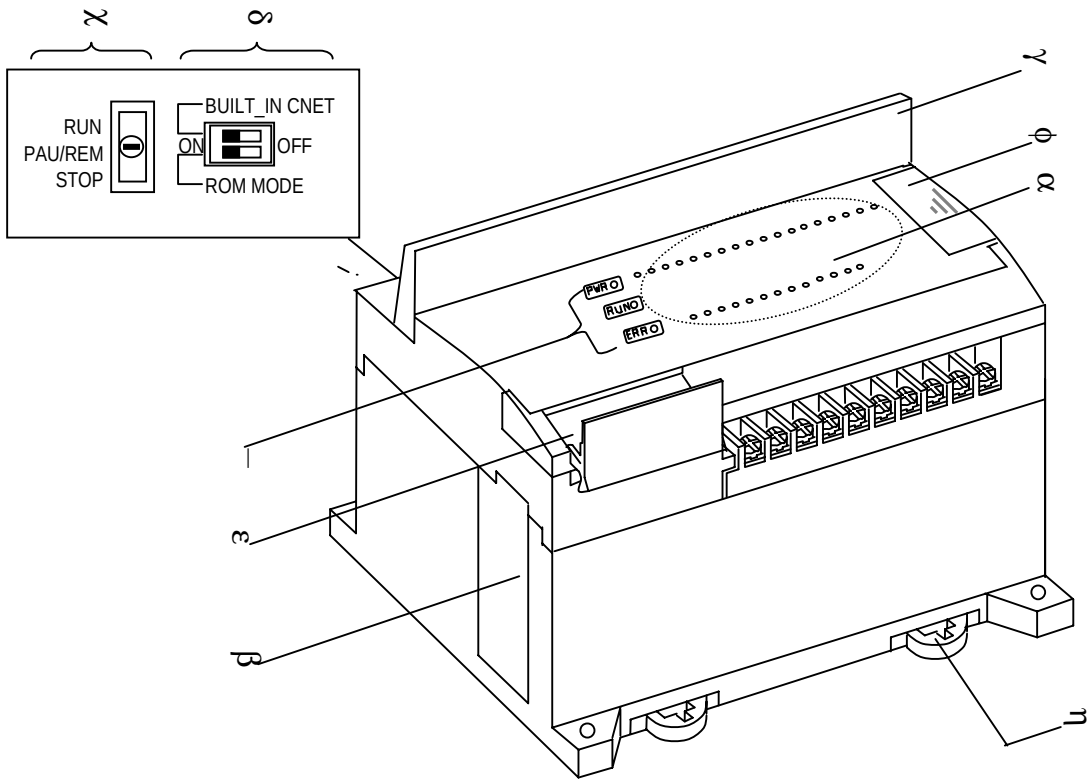
<table>
<tr><th>No.</th><th>Item</th><th colspan="6">Specifications</th><th>References</th></tr>
<tr><td>1</td><td>Operating ambient Temperature</td><td colspan="6">0 ~ 55 °C</td><td></td></tr>
<tr><td>2</td><td>Storage ambient Temperature</td><td colspan="6">–25 ~ +70 °C</td><td></td></tr>
<tr><td>3</td><td>Operating ambient Humidity</td><td colspan="6">5 ~ 95%RH, non-condensing</td><td></td></tr>
<tr><td>4</td><td>Storage ambient Humidity</td><td colspan="6">5 ~ 95%RH, non-condensing</td><td></td></tr>
<tr><td rowspan="8">5</td><td rowspan="8">Vibrations</td><td colspan="4">Occasional vibration</td><td colspan="2">-</td><td></td></tr>
<tr><td>Frequency</td><td colspan="2">Acceleration</td><td>Amplitude</td><td colspan="2">Sweep count</td><td rowspan="7">IEC 61131-2</td></tr>
<tr><td>10≤f＜57Hz</td><td colspan="2">–</td><td>0.075mm</td><td colspan="2" rowspan="5">10 times for each X, Y, Z axis</td></tr>
<tr><td>57≤f≤150Hz</td><td colspan="2">9.8m/s^2 {1G}</td><td>–</td></tr>
<tr><td colspan="4">Continuous vibration</td></tr>
<tr><td>Frequency</td><td colspan="2">Acceleration</td><td>Amplitude</td></tr>
<tr><td>10≤f＜57Hz</td><td colspan="2">–</td><td>0.035mm</td></tr>
<tr><td>57≤f≤150Hz</td><td colspan="2">4.9m/s^2 {0.5G}</td><td>–</td></tr>
<tr><td>6</td><td>Shocks</td><td colspan="6">• Maximum shock acceleration: 147 m/s^2 {15G}
• Duration time: 11ms
• Pulse wave: half sine pulse (3 shocks per axis, on X, Y, Z axis)</td><td>IEC 61131-2</td></tr>
<tr><td rowspan="5">7</td><td rowspan="5">Noise Immunity</td><td>Square wave Impulse noise</td><td colspan="5">±1,500 V</td><td>IMOIS' Internal Standard</td></tr>
<tr><td>Electronic discharge</td><td colspan="5">Voltage: 4 kV (Discharge by contact)</td><td>IEC 61131-2, IEC 801-2</td></tr>
<tr><td>Radiated electromagnetic field noise</td><td colspan="5">27 ~ 500 MHz, 10 V/m</td><td>IEC 61131-2, IEC 801-3</td></tr>
<tr><td rowspan="2">Fast transient /burst noise</td><td>Item</td><td>Power supply</td><td>Digital I/O (>24V)</td><td colspan="2">Digital I/O (<24V) Analog I/O Interface</td><td rowspan="2">IEC 61131-2 IEC 801-4</td></tr>
<tr><td>Voltage</td><td>2kV</td><td>1kV</td><td colspan="2">0.25kV</td></tr>
<tr><td>8</td><td>Atmosphere</td><td colspan="6">Free of corrosive gases and excessive dust</td><td>IEC61131-2</td></tr>
<tr><td>9</td><td>Altitude</td><td colspan="6">Up to 2,000m</td><td></td></tr>
<tr><td>10</td><td>Pollution degree</td><td colspan="6">2</td><td></td></tr>
<tr><td>11</td><td>Cooling method</td><td colspan="6">Air-cooling</td><td></td></tr>
</table>

REMARK

1) IEC (International Electrotechanical Commission): An international civilian institute who establishes international standards in area of electrics and electronics.
2) Pollution degree: An indicator, which indicates pollution degree, which determine insulation performance of equipment.
Pollution degree 2 : Normally, only non-conductive pollution occurs. Occasionally, however, a temporary conductivity caused by condensation shall be expected.

Chapter 4. Names of Parts

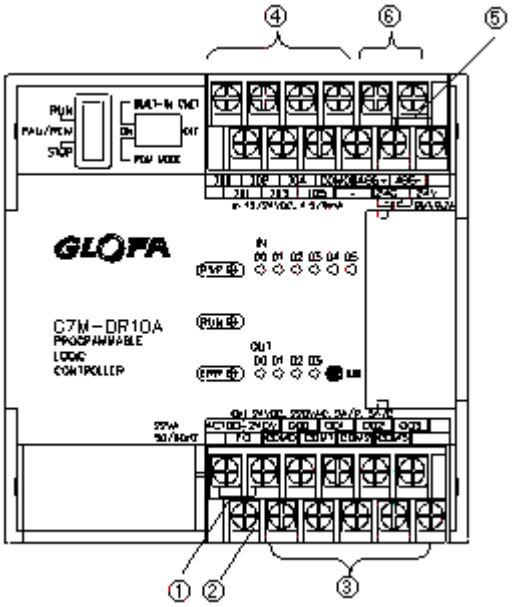
4.1 Base Unit



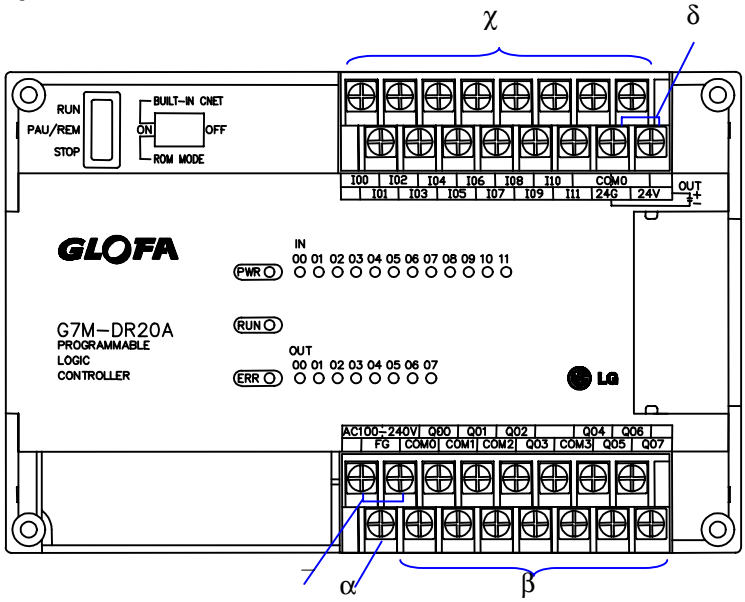
No	Name		
1	CPU Condition LED Indication	PWR LED	Indicates power supply to the system • On: When the supply is normal • Off: When the supply is abnormal
		RUN LED	Indicates base unit operation • On: Indicates local key switch or remote running mode • Off: with the following led gets off Without normal power supply to the base unit While key switch is stopped Detecting an error makes operation stop
		ERR LED	Indicates Base Units operation • On/Off of led: self-inspected error • Off: CPU is normally working.
2	I/O LED		Indicates I/O operating status
3	Folder for battery installation		Folder for back-up battery installation

No	Name	
4	Key switch mode creation	Indicates base units drive mode • RUN: Indicates program operation • STOP: Stopped program operation • PAU / REM: usage of each modules are as follows: PAUSE : temporary stopping program operation REMOTE : Indicates remote drive
5	Dip-switch memory operation	See Chapter 5
6	RS-232C connector	9-pin DIN connector to connect with external devices like GMWIN
7	Expansion connector cover	Connector cover to connect with expansion unit
8	Terminal block cover	Protection cover for wiring of terminal block
9	Private hook DIN rail	Private part hook for DIN rail

4.1.1 10-point base unit

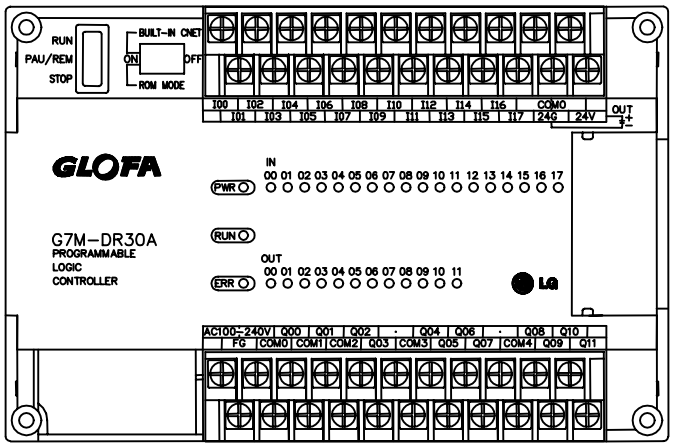


4.1.2 20-point base unit

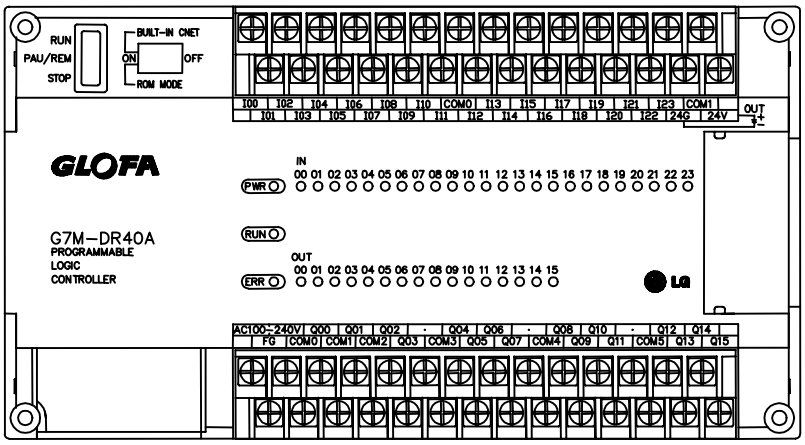


No.	Name	Usage
1	Terminal block for power supply	Terminal blocks for power supply (AC 100V ~ 240V or DC12 ~ 24V))
2	FG circuit	Frame ground (AC power model)
	LG circuit	Line ground (DC power model)
3	Output circuit	Output connecting current
4	Input circuit	Connecting input current
5	DC24V, 24G circuit	Service power supply for DC 24V needed place
6	RS-485 communication terminal	Terminal block for RS-485 communication (10-point base unit only)

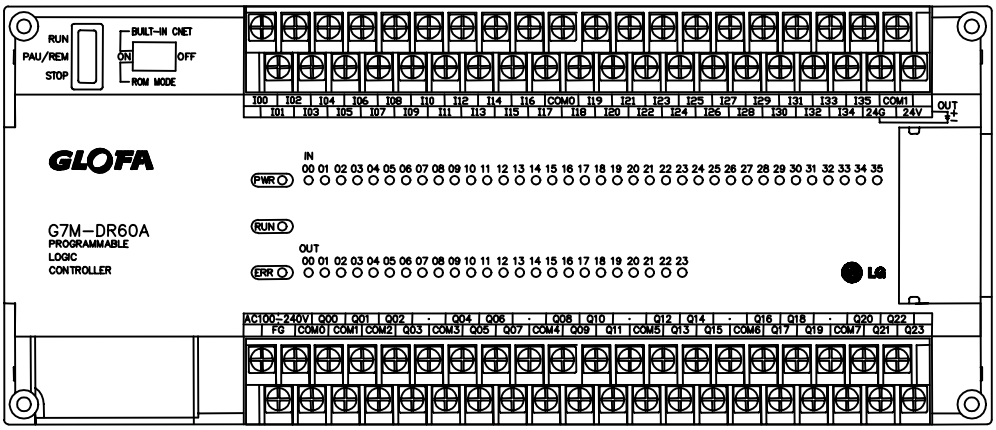
4.1.3 30-points base unit



4.1.4 40-points base unit

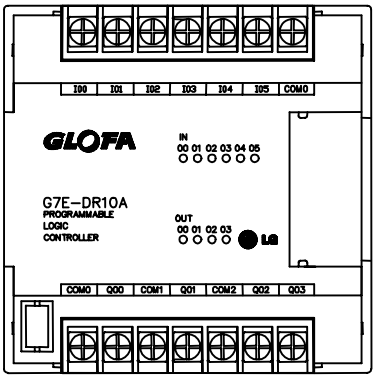


4.1.5 60-points base unit

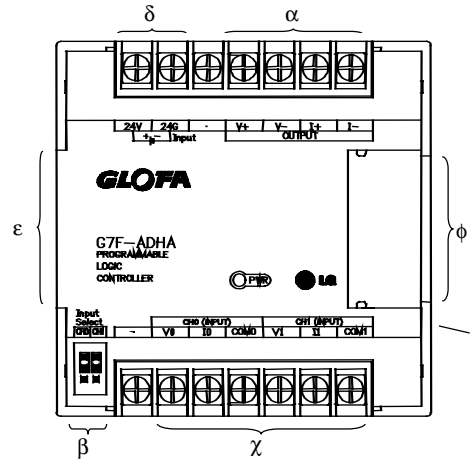


4.2 Expansion Module

4.2.1 Digital I/O Module

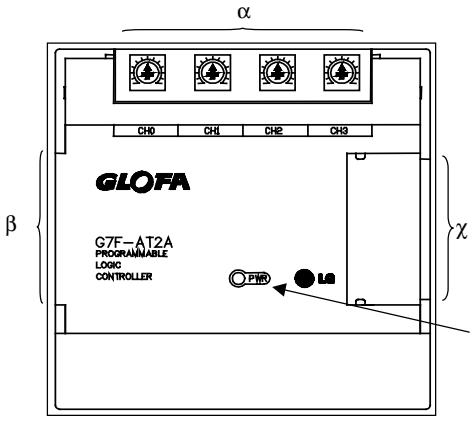


4.2.2 A/D · D/A Combination Module



No.	Names
—	RUN LED
α	Analog Input Terminal
β	Analog Input (Voltage/current) selecting jumper pin
χ	Analog Output Terminal
δ	External Power Supply Terminal (DC24V)
ε	Expansion Cable
φ	Expansion Cable Connecting Terminal

4.2.3 Analog Timer Module



No.	Names
—	RUN LED
α	Analog Timer Volume Control Resistance
β	Expansion Cable
χ	Expansion Cable Connecting Terminal

4.2.4 Option Module

GM7 series have two types of option modules. These modules are attached to the expansion slot of main or expansion unit.

No	Name
1	Option module
2	Connector

Chapter 5. CPU

5.1 Power Supply Specifications

5.1.1 AC power supply

Item \ Model		G7M–DR10A G7M–DT10A	G7M–DR20A G7M–DT20A	G7M–DR30A G7M–DT30A	G7M–DR40A G7M–DT40A	G7M–DR60A G7M–DT60A
Input	Voltage	AC85 ~ 264V				
	Frequency	50 / 60 Hz (47 ~ 63 Hz)				
	Current	0.5A(AC110V)/0.25A(AC220V)			0.6A(AC110V)/0.3A(AC220V)	
	Inrush current	30 A or less				
	Efficiency	65% or higher (rated input/load)				
	Fuse	1A/Slow/AC250V			2A/Slow/AC250V	
	Dropout tolerance	20 ms or less				
Output (1)	Voltage	DC 5V	DC 5V		DC 5V	
	Current	1.0 A	1.2 A		2.0 A	
Output (2)	Voltage	DC24V				
	Current	0.2 A				
Output status indication		PWR LED On when normal output status				

5.1.2 DC power supply

Item \ Model		G7M–DR10A/DC	G7M–DR20A/DC	G7M–DR30A/DC	G7M–DR40A/DC	G7M–DR60A/DC
Input	Voltage	DC10.2 ~ 28.8V				
	Current	1.2A(DC12V) / 0.6A(DC24V)			1.8A(DC12V) / 0.6A(DC24V)	
	Inrush current	60 A or less			70 A or less	
	Efficiency	60% or higher (rated input/load)				
	Fuse	5A/Slow/50V				
	Dropout tolerance	1 ms or less				
Output (1)	Voltage	DC 5V	DC 5V		DC 5V	
	Current	1.0 A	1.2 A		2.0 A	
Output status indication		PWR LED On at normal output status				

5.2 CPU Specifications

The following table shows the general specifications of the IMO-GM7 series

Items		Specifications	Remarks
Operation method		Cyclic operation of stored program, Interrupt task operation	
I/O control method		Scan synchronized batch processing method (Refresh method)	Immediate input/output is available by 'direct I/O' function
Programming language		Instruction List Ladder Diagram Sequential Function Chart	
Number of instructions	Operator	LD: 13, IL: 21	
	Basic function	138	
	Basic function block	11	
	Special function block	Each special module have their own special function blocks	
Processing speed	Operator	0.5	
	Basic function	Refer to Appendix 3	
	Basic function block		
Programming memory capacity		68K bytes	Including parameter (Approx. 4k byte)
I/O points		10 points expansion unit : input 6 points/output 4 points 20 points base unit : input 12points/output 8point 30 points base unit : input 18points/output 12point 40 points base unit : input 24points/output 16point 60 points base unit : input 36points/output 24point	Max 2 expansion units can be attached to a base unit
Data memory	Direct variable area	2k to 8k bytes	Adjustable with parameter setting
	Symbolic variable area	32 k bytes-Direct variable area	
Timer		No limitations in points Time range:0. 001 to 4294967.295 sec (1193 hours)	1point occupies 20 bytes of Symbolic variable area
Counter		No Limitations in points Counting range: -32768 to +32767	1point occupies 8bytes of symbolic variable area
Operation modes		RUN, STOP, PAUSE and DEBUG	
Data protection method at power failure		Set to 'Retain' variables at data declaration	
Number of program blocks		128	

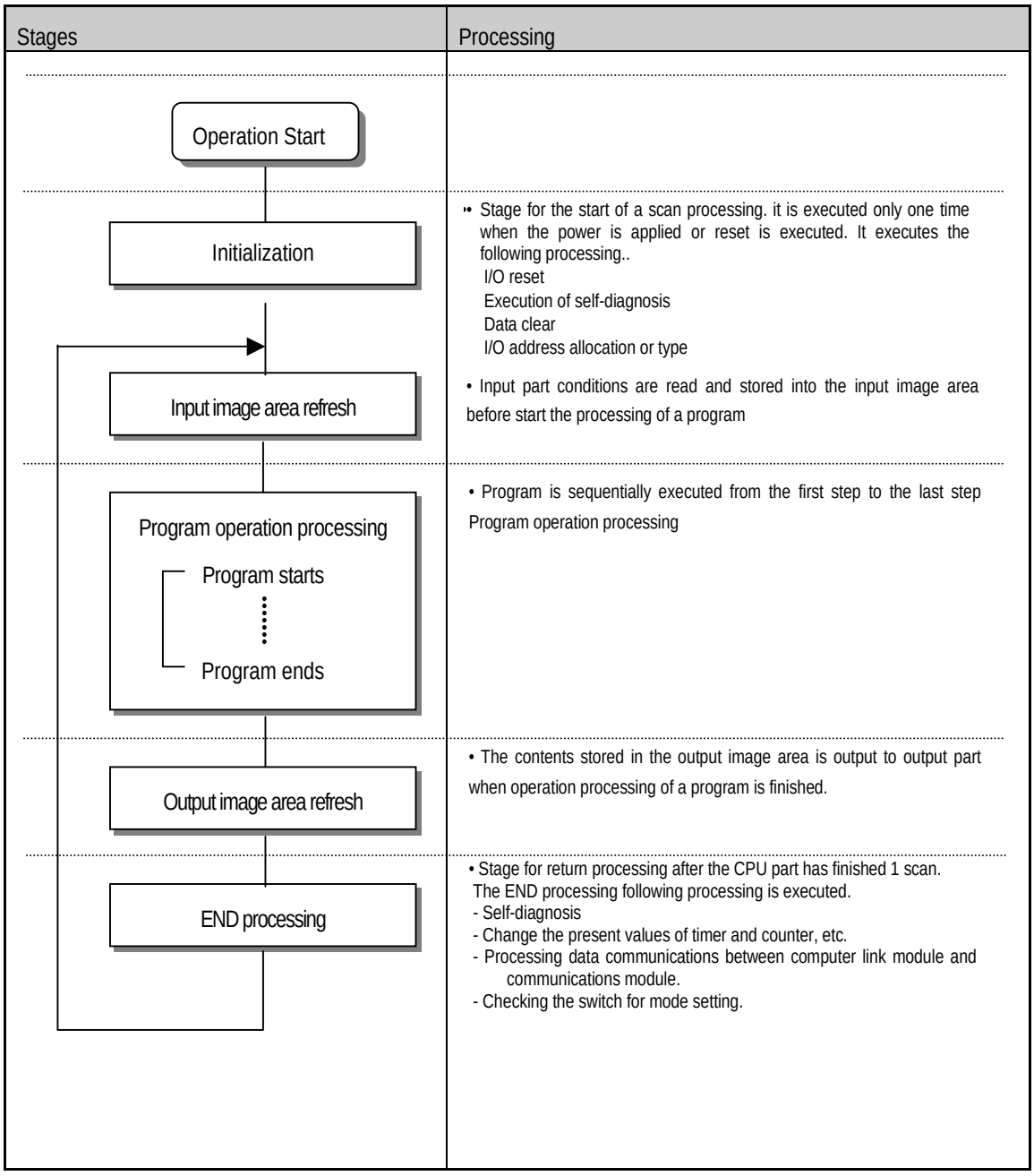
Items			Specifications	Remarks
Program type	Scan		100	
	Time-driven interrupt task		8	
	External interrupt task		8 Total 8 pieces are usable.	
	High speed counter task		1	
	Inside interrupt task		8	
	Initialization task		1 (_INIT)	
Internal Function	PID control function		Function block control, auto tuning, forced output, adjustable operation scan time, forward/reverse operation control	
	Cnet I/F Function		IMO exclusive protocol support MODBUS protocol support User's protocol support	Common use with GMWIN port
	High-speed counter	Capacity	1 phase : 16 kHz, 1 channel 2 phase : 8 kHz,1 channel	
		Counter function	It has 3diffferant counter function as following; 1 phase, up/down by program 1 phase, up/down by B phase input 2 phase, up/down by phase difference	
		Multiplication function	Multiplication : 1, 2, or 4 (adjustable)	
		Data comparison function	Execute a task program when the elapsed counter value reaches to the preset value	
	Pulse catch		Minimum pulse width : 0.2msec, 8 points	
	Pulse output		2khz, 1point	Transistor output only
	External interrupt		8points, 0.4ms	
	Input filter		0~15ms	
Weight (g)	G7M-DR20A		480	
	G7M-DR30A		551	
	G7M-DR40A		670	
	G7M-DR60A		844	
	G7E-DR10A		228	

5.3 Operation Processing

5.3.1 Operation Processing Method

1) Cyclic operation

A PLC program is sequentially executed from the first step to the last step, which is called scan. This sequential processing is called cyclic operation. Cyclic operation of the PLC continues as long as conditions do not change for interrupt processing during program execution. This processing is classified into the following stages:


Stages
Processing
Operation Start
Initialization
Stage for the start of a scan processing. it is executed only one time when the power is applied or reset is executed. It executes the following processing..
I/O reset
Execution of self-diagnosis
Data clear
I/O address allocation or type
Input image area refresh
• Input part conditions are read and stored into the input image area before start the processing of a program
Program operation processing
Program starts
Program ends
• Program is sequentially executed from the first step to the last step Program operation processing
Output image area refresh
• The contents stored in the output image area is output to output part when operation processing of a program is finished.
END processing
• Stage for return processing after the CPU part has finished 1 scan.
The END processing following processing is executed.
- Self-diagnosis
- Change the present values of timer and counter, etc.
- Processing data communications between computer link module and communications module.
- Checking the switch for mode setting.

2) Time driven interrupt operation method

In time driven interrupt operation method, operations are processed not repeatedly but at every pre-set interval. Interval, in the GM7 series, can be set to between 0.001 to 4294967.29 sec. This operation is used to process operation with a constant cycle.

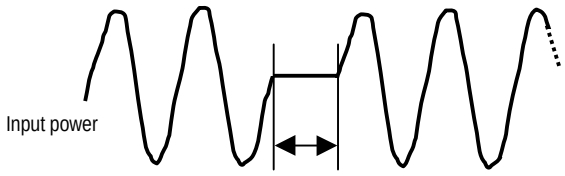
3) Event driven interrupt operation method

If a situation occurs which is requested to be urgently processed during execution of a PLC program, this operation method processes immediately the operation, which corresponds to interrupt program. The signal, which informs the CPU of those urgent conditions is called interrupt signal. The GM7 CPU has two kind of interrupt operation methods, which are internal and external interrupt signal methods.

5.3.2 Operation processing at momentary power failure occurrence

The CPU detects any momentary power failure when the input line voltage to the power supply falls down below the defined value. When the CPU detects any momentary power failure, the following operations will be executed:

1) Momentary power failure within 20 ms



Momentary power failure exceeding 2Oms

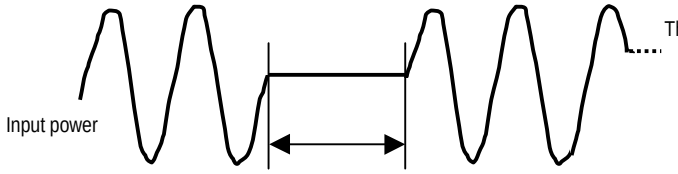
(1) The operation processing is stopped with the output retained.

(2) The operation processing is resumed when normal status is restored.

(3) The output voltage of the power supply retains the defined value.

(4) The watchdog timer (WDT) keeps timing and interrupt timing normally while the operations is at a stop.

2) Momentary power failure exceeding 20 ms



The re-start processing is executed as the power is applied.

Momentary power failure exceeding 2Oms

REMARK

1) Momentary power failure

The PLC defining power failure is a state that the voltage of power has been lowered outside the allowable variation range of it. The momentary power failure is a power failure of short interval (several to tens ms).

5.3.3 Scan Time

The processing time from a 0 step to the next 0 step is called scan time.

1) Expression for scan time

Scan time is the addition value of the processing time of scan program that the user has written, of the task program processing time and the PLC internal processing time.

(1) Scan time = Scan program processing time + Task program processing time + PLC internal processing time

• Scan program processing time = The processing time used to process a user program that is not specified to a task program.

• Task program processing time = Total of the processing times of task programs executed during one scan.

• PLC internal processing time = Self-diagnosis time + I/O refresh time + Internal data processing time + Communications service processing time

(2) Scan time differs in accordance with the execution or non-execution of task programs and communications processing, etc.

2) Flag

Scan time is stored in the following system flag area.

- _SCAN_MAX: Maximum scan time (unit: 1 ms)
- _SCAN_MIN: Minimum scan time (unit: 1 ms)
- _SCAN_CUR: Current scan time (unit: 1 ms)

5.3.4 Scan Watchdog Timer

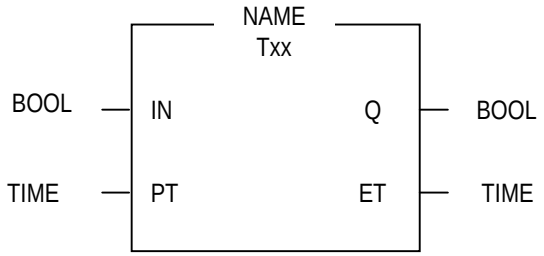
1) Watchdog timer is used to detect a delay of abnormal operation of sequence program (Watchdog time is set in menu of basic parameter of GMWIN.)
2) When watchdog timer detects an exceeding of preset watchdog time, the operation of PLC is stopped immediately and all output is off.
3) If an exceeding of preset watchdog time is expected in sequence program, use 'WDT_RST' function. 'WDT_RST' function make elapsed watchdog time as zero.
4) In order to clear watchdog error, using manual reset switch, restarting the PLC or mode change to STOP mode are available.

REMARK

Setting range of watchdog : 1~ 5,000ms(unit : 1ms)

5.3.5 Timer Processing

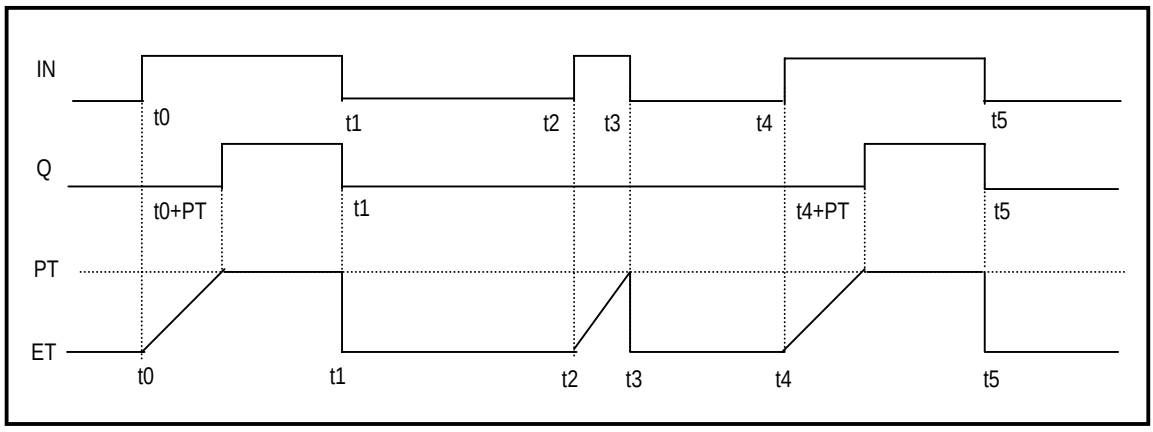
The CPU timer is an incremental timer, which increases its present value according to the measuring time. Three types of On Delay Timer (TON), Off Delay Timer (TOF) and Pulse Timer (TP) are available. Its measuring range is 0.001 to 4,294,967,295 sec (1,193 hours) by 1 ms. for details, refer to “IMO-GM programming”.



1) On Delay Timer : Process Time Change and Contact On/Off

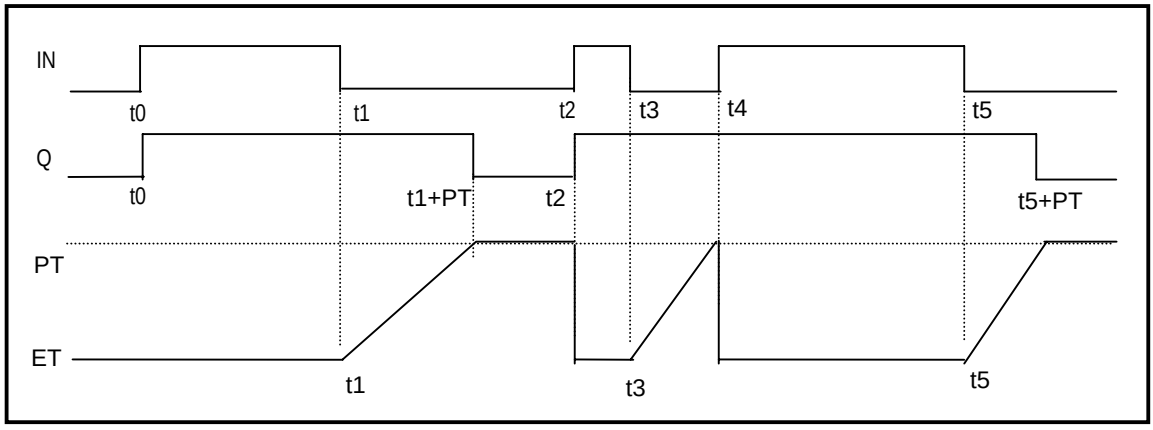
Timer Process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the Timer output contact turns on.

On Delay Timer Timing Diagram is shown as below.



2) Off Delay Timer : Process Time Change and Contact On/Off

- If input condition turns on, timer output contact (Q) turns on. If input condition turns off, timer process time starts increasing.
- The process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the contact (Q) turns off. The following diagram shows Off Delay Timer Timing.



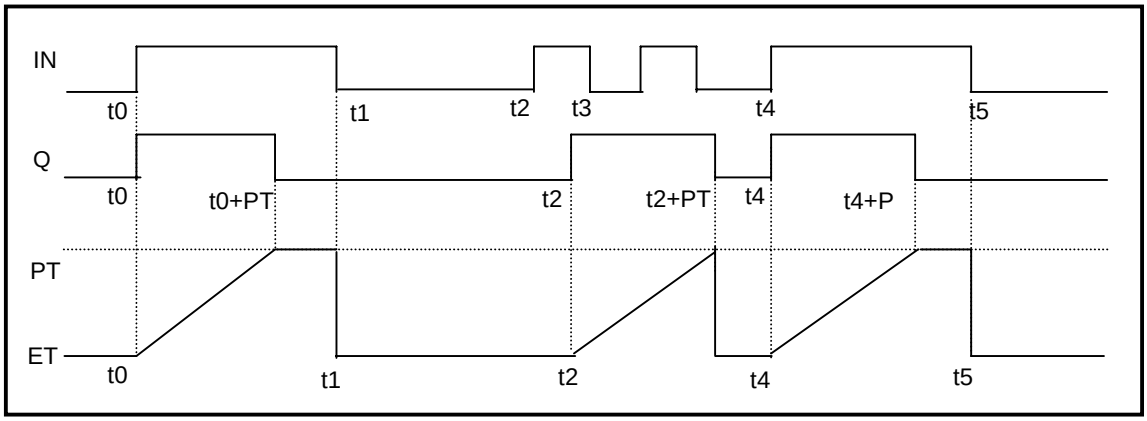
3) Pulse Timer Process Time Change and Contact On/Off

If input condition turns on, output contact (Q) turns on.

The process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the contact (Q) turns off.

The contact turns off after the setting time regardless of input condition off status.

The following diagram shows pulse timer timing.



4) Timer Error

The maximum timer error is '1 scan time + time from the start of scan to execution of the timer function block'

5.3.6 Counter Processing

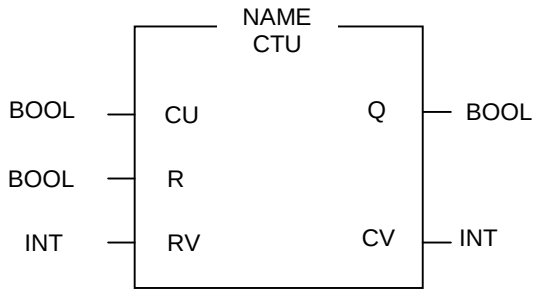
The CPU part counter increase/decrease the present counting value by the detection of rising edge (off on) of input signal. Three types of counter are increment counter, Decrement counter and Increment/Decrement Counter. For details, refer to IMO — GM Programming'.

- The Increment counter is a counter which increment the present counting value
- The Decrement counter is a counter which decrement the present counting value
- The Increment-Decrement counter is a counter, which compares the counting values of two input conditions.

1) Counter Present Value Change and Contact On/Off

Increment Counter

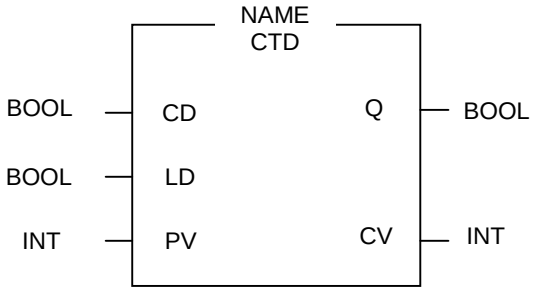
- It should have Input condition (CU), reset condition (R) and setting value (PV).



- If the counting value (CV) increments and reaches the setting value (PV), the output contact (Q) turns on. When the reset signal is turn on, the counting value is set to 0' and the output contact (Q) turns off.

(2) Decrement Counter

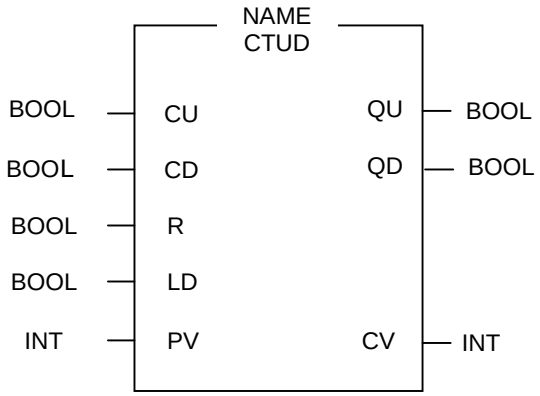
- It should have input condition (CD), load (LD) and setting value (PV).



- If the counting value (CV) decrements and reaches 0', the output contact (Q) turns on. If the load (LD) signal is turned on, the counting value is set to the setting value and the output contact (Q) turns off.

Increment/Decrement Counter

- It should have Increment input condition (CU); Decrement input condition (CD), load (LD) and setting value (PV).



- If reset signal (R) turns on, counting value (CV) is set to 0.
- If load signal (LD) turns on; counting value is set to setting value (PV).
- It is increased by 1 at the rising edge of increment input (CU) and decreased by 1 at the edge of decrement input (CD). If counting value (CV) is equal or larger than setting value (PV), QU will be on, and if counting value (CV) is equal or less than setting value (PV), QD will be on.

2) Counting speed

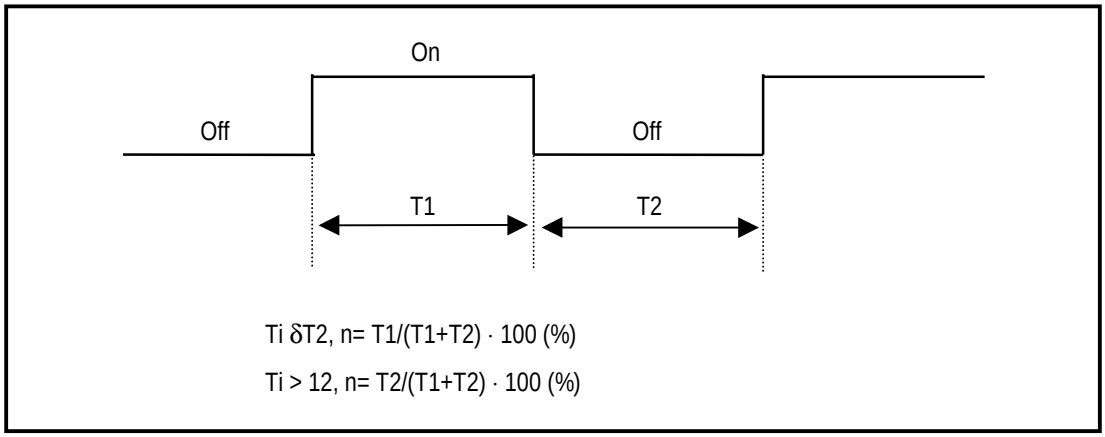
- The counting speed is decided by scan time and it will be counted when on time or off time of input condition is larger than each scan time.

Max. Counting speed Cmax = N/100 * 1/ts (pps/s)

n: duty (%)

ts: scan time(s)

- Duty (n) is the percentage (%) of On/Off of the input signal.



5.4 Program

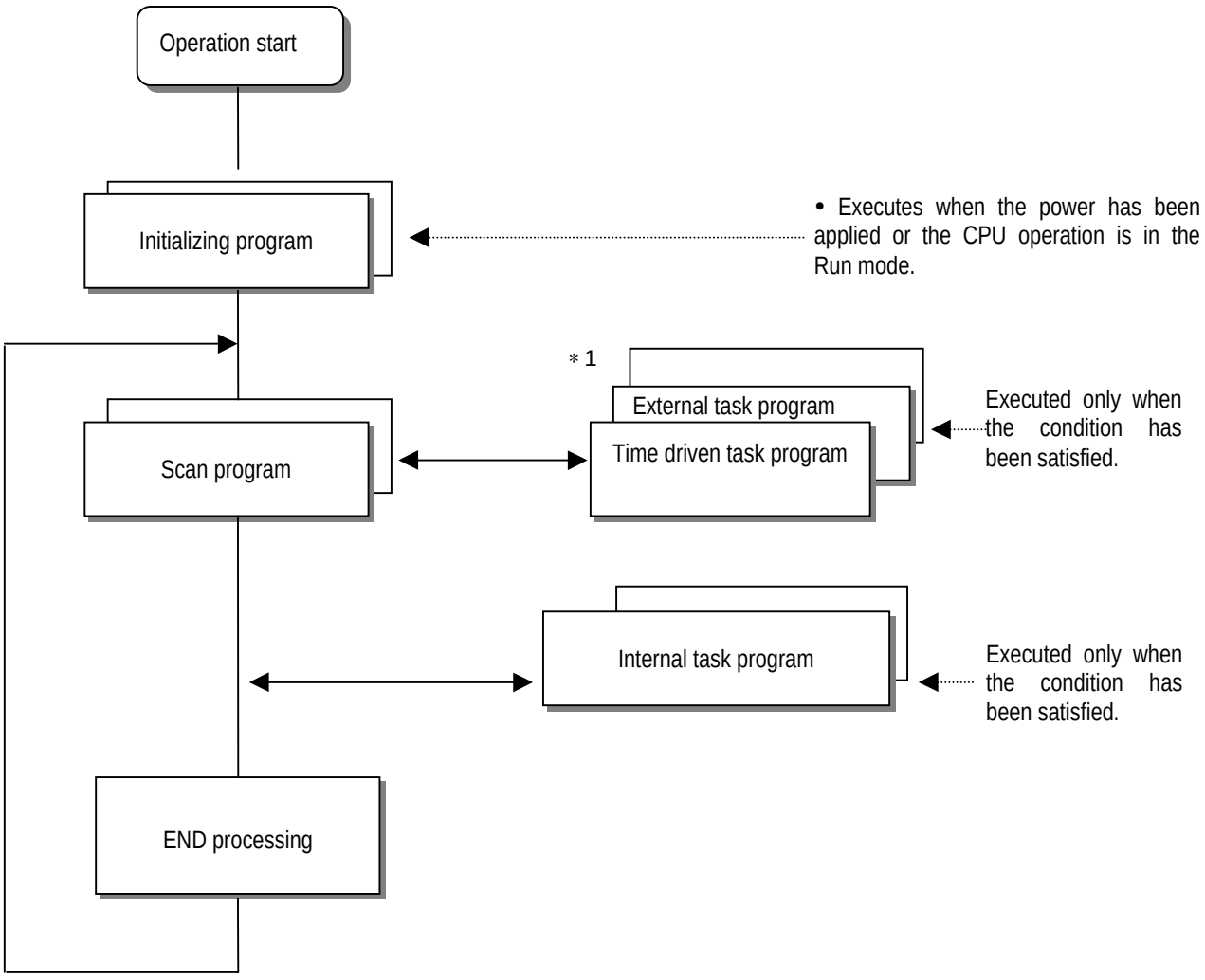
5.4.1 Program Configuration

A program consists of all of the function elements that are needed to execute a particular control. It is to be stored in the internal RAM of the CPU part or the flash memory. The function elements are classified as below.

Function elements	Processing Operation
Initialization program	• Executed when the power is applied or the CPU operation is transited to the RUN mode. • Executes the initial/fixes data setting for execution of scan program and the initialization of peripheral devices on special modules.
Scan program	• Processes the constantly repeated signals which are executed every scan.
Time driven task Program	• When the following time conditional processing is required the program is executed complying with the time interval setting. ∂ In case of the processing need a shorter interval than that of average scan processing time. ∂ In case of the processing needs a longer interval than that of average scan processing time. ∂ In case that the processing should be executed by the specified time interval.
Event driven task Program	• A shorter processing is executed for internal or external interrupt.

5.4.2 Program Execution Procedure

The followings explain the program execution procedure when the power is applied or the mode-setting switch of CPU part is in the RUN status. Program operation processing is executed as the procedure given below:


Operation start
Initializing program
• Executes when the power has been applied or the CPU operation is in the Run mode.
∗1
External task program
Time driven task program
Executed only when the condition has been satisfied.
Scan program
Internal task program
Executed only when the condition has been satisfied.
END processing

REMARK

∗ 1: In the IMO PLC, the time driven interrupt task programs and event driven interrupt task programs are called task program. Event driven programs are classified into single task (internal interrupt) or interrupt task (external interrupt) according to the S/W and H/W interrupt signaling method.

1) Initialization program

(1) Function

- The Initialization program initializes the program to execute scan and task programs.

(2) Cold/warm restart program

- The initialization program specified to _INIT task is executed with cold or warm restart mode when the operation starts.
- This initialization program executes the operations repeatedly until the setting conditions are satisfied (that is, until the Flag _INIT_DONE in the initialization program turns on). However, the I/O refresh is still executed.

(3) Flag

- _INIT_RUN flag is on during executing the initialization program.

2) Scan program

(1) Function

- In order to process signal, which repeats constantly, the program executes its sequential operation repeatedly from the first step to the end step.
- If the interrupt task execution condition has been satisfied by a time driven task or event driven task during scan program execution, the program that is under execution will be temporary stopped and the corresponding task program will be executed.

(3) Configuration

- Up to 100 scan programs can be used.
  (If task programs are used, the usable number is reduced as many as that of the used task programs)
- Program has been not specified to initialization or task program when writing that program, it will be automatically specified to scan program.
- Scan program has lowest execution priority and the priorities of scan program are determined their registration sequence in the GMWIN screen when writing those programs.

3) Task program

(1) Function

- In order to process internal/external signal, which occurs periodically, or non-periodicity the task program temporarily stop the operation of scan program and processes first the corresponding function.

(2) Types

- Task programs are classified into the four types as below.

  Time driven task program : Up to 8 programs are applicable

  Single (internal) task program: Up to 8 programs are applicable

  Interrupt (external) task program: Up to 8 programs are applicable

  High speed counter task program: only 1 program is applicable.

- Time driven task program

  The program is executed by the time internal set before.

- Single (internal) task program

  The corresponding program will be executed at the rising edge and on state of internal contact in the program.

  The detection of the start up condition will be executed after the scan program has been processed.

- Interrupt (external) task program

  The program is executed according to the external signal a input to the interrupt module

- High-speed counter task program

  The program is executed according to speed level.

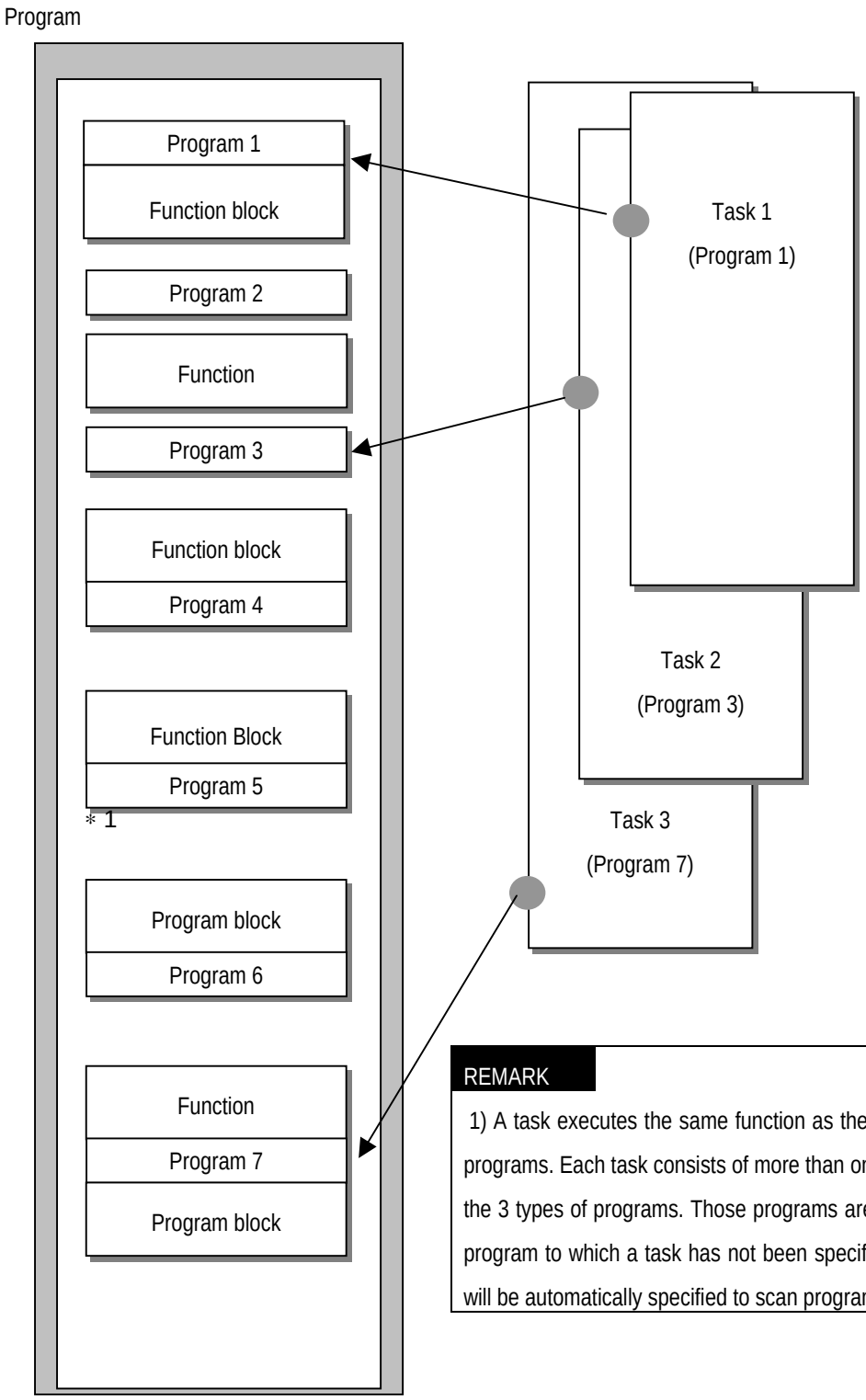
Remark

1) Refer to section 5.3.3 “Task” for details of task program.

2) GM7 series uses separate input program to manage sign of interrupt. Refer to section 5.3.3 “Task” for details of task program.

5.4.3 Task

The followings explain the program structure and tasks of the GMWIN, that is, the IMO-GM programming S/W, in order to give an understanding of the task function. (Refer to GIMWIN section for details of GMWIN program)



REMARK

1) A task executes the same function as the control panel executing programs. Each task consists of more than one program blocks out of the 3 types of programs. Those programs are called task program. A program to which a task has not been specified as marked with "*1", will be automatically specified to scan program

1) Task types and functions

The following table shows the types and functions of tasks.

Type / Size	Time driven task	External interrupt task	Internal interrupt task	High speed task
Number	8	8	8	1
Start up condition	Time driven interrupt (up to 4,294,967.29 sec by the 10 ms)	At the rising edge of input contact on the designated slot.	The rising edge or on state of the BOOL variable data which has been specified of buffer data.	Using CHSC_SET F/B, select a set value.
Detection and execution	Executed periodically as setting time	Immediately executed when an edge occurs in the interrupt module	Executed with edge detection after scan program has been finished.	When reaches the SV, it executes.
Detection delay time	Up to 1 ms delay	0.4ms	Delayed for the same time as max. scan time.	Delayed up to 1 ms.
Execution priority	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7 (Level 0 has highest priority)

2) Task program processing Method

The following explains the common processing method and instructions for task programs

(1) Task program characteristics

- The task program will be executed when an execution condition is satisfied while the scan program is repeatedly processed at every scan. Be sure to consider that point when writing a task program
- For example, if a timer and a counter have been used in a 10 sec cycle time driven task program, the timer can occur up to 10 sec error and an input which has been changed within 10 sec will not be counted because the counter checks its input status every 10 sec

(2) Execution priority

- The higher priority task program will be executed firstly.
- If a newly invoked task has higher priority than that of existing tasks which are under execution, they are temporary stopped and task has higher priority will be executed.
- When determining the priority of a task program, consider the characteristics, importance and urgency of the program

REMARK

The priority for GM7 can't be set as the same. If it is set as the same, an error will occur.

(3) Processing delay time

The following factors influence on the processing delay of task program, consider the characteristics, importance and urgency of the program.

- Task detection delay (Refer to the detailed description of each task)
- Execution delay due to the execution of prior task programs
- Delay due to the execution of higher priority task programs white executing task programs

(4) Relationship of task program to initialization or scan program

- User defined tasks will not start while the initialization task program is being executed.
- As scan program has the lowest priority, if a task is invoked the scan program will be stopped and the task programs will be processed prior to them. Therefore, if tasks are invoked many times or concentrated sometimes the scan time may be extended abnormally. Be cautious when setting task conditions.

(5) Protection of the programs under execution from task programs

- If problems can be occur in case that program lose its execution continuousness by the task programs which have higher proprieties, the execution of task programs can be partly perverted For program protection, use the DI function (Task program start-up disable) or EI function (task program start-up enable)
- Use 'DI' function where program needs protection and 'EI' function where program needs cancellation. After the scan program ends of the running program, automatically it becomes permissible. Initialization program doesn't get influences from 'DI and EI.'

3) Time driven task program processing method

The followings explain the processing method of a task program when its task condition (start-up condition) has been set to drive by time.

(1) Settings that have to be set for the task

- Set the task execution cycle and its priority, which are used as start-up conditions for the task programs to be executed. Priority number will be task number.

(2) Time driven task processing

- The corresponding time driven interrupt task program will be executed every setting time internal (execution cycle).

(3) Precautions for using the time driven task program

- While a time driven task program is being executed or ready for its execution, if a same priority task program has been invoked to be executed the newly invoked task will be ignored, the representative task collision warning flag (TASKERR) will be set to ON, the detailed system error flag (JC BMAP[n] will be set to ON at its corresponding location and occurrence time of the time driven tasks whose execution requests have been ignored will be written at its corresponding location of the flag TC_CNT[n].
- The timer that invokes the execution request for time driven task programs will be incremented only when the operation mode is in the RUN mode

- If the RUN mode has been changed into the PAUSE mode while operating with the RUN mode, and then the operation mode has been changed again into the RUN mode, the operation time spent with the PAUSE mode will be ignored.
- When setting the execution cycle for a time driven task program, be cautious that execution requests for many time driven task programs can occur. If four time driven task programs of cycle 2, 4,10 and 20sec are used, four execution requests will occur every 20 sec and scan time can be momentarily extended.

4) External contact program processing method

In GM7series, it is different from GM1/2/3/4 to use normal digital input task program, not a separate interrupt input module. The following explains in the case that the task(start-up condition) of a task program has been set to an external input signal.

(1) Settings that have to be set for the task

- Set the contact No. of input module and priority for the task that will be used as start-up conditions of the task programs to be executed. Priority will be the task number.

(2) External contact task processing

- The CPU module checks the occurrence of interrupt input every lms and executes the task program, which are designated by the contact at which the signal has been occurred.

(3) Precautions for using an external contact task.

- Input interrupt that is possible to set is up to %IX0.0.0~%IX0.0.7.
- While a task program which are designated by an input module having interrupt input, contact is being executed or ready for its execution, if an execution request of a task program has been occurred to the same input contact then the newly invoked task will be ignored, the representative task collision warning flag(_TASK_ERR) will be set to ON, the detailed system error flag(_TC_BAMP[n], TC_CNT[n] will be set to ON at its corresponding location and the occurrence time of the external task whose execution request has been congested.
- Execution request for a task program can be accepted only when the operation mode is in the RUN mode. That is, if the RUN mode has been changed into the PAUSE mode while operating with the RUN mode and the operation mode has been changed into the RUN mode again, all execution requests occurred during the operation with the PAUSE mode will be ignored.

5) Internal task program processing method

The following explains the processing method when the task (start-up condition) of a task program has been set to the contact of direct variable area(I, Q or M) or automatic variable area.

(1) Settings that have to be set for the task.

- Set the contact No. of input module and priority for the task that will be used as start-up conditions of the task programs to be executed. Priority will be the task number.

(2) Internal contact task processing

- After the execution of scan program has been completed in the CPU module, the internal contacts that are the start-up conditions of the task program will be checked and the internal task programs where rising edge or on state has been occurred will be executed in accordance with its parameter.

(3) Precautions when using an internal task program.

- The internal task program is executed when scan program has finished its execution. Therefore, though the execution condition for the internal task program has been invoked in the scan program or task program(time driven, external) the task (start-up condition) will not be immediately executed but will be executed when scans program has finished its execution.
- If execution of an internal task program is requested, the execution conditions will be checked when scan program has finished its execution. Therefore, if an internal task execution conditions, during 'One' scan, has been occurred and disappeared (if the specified contact has been turned from OFF to ON, and then from ON to OFF) by scan program or (time driven or external) task program the task will not be executed as the execution condition can not be detected at the time that execution conditions are being checked.

REMARK

1) When an action must continuously be executed according to the related contact point set as a start-up condition, select a level.

6) Execution of high-speed task program

GM7 series uses general digital input contact point to count high-speed pulse, not a separate high-speed pulse input module. Setting a task (startup condition) as the same with the one of the high-speed pulse input will be explained.

(1) Conditions to be set for a task

- Set the priority on the tasks that are startup conditions for the task program to be executed. Then a task number will automatically be added in the priority order.

(2) Processing the high speed counter task

- When CHSC_SET F/B of the program assigns a set value, the task program whose set value matches with the counted value of the pulse that is input in a high speed is executed.

(3) Precautions for using high speed counter task program

- The task can be used only with CHSC_SET F/B.
- High speed input counter can be used without CHSC_SET F/B.
- Even though the operation is PAUSE mode, counted value rises. But this can executes the task program. When the operation is RUN mode, the task is executed.

7) Examination on task program

After writing down a task program, be sure to examine the following items.

(1) Task setting has been correctly done?

- If tasks are invoked more frequently than necessary or several tasks are invoked simultaneously within one scan, the scan time become longer and irregular. If the task setting cannot be changed, check the maximum scans time.

(2) Task priorities are properly arranged?

- The lower priority tasks still may not be processed after its time due to delay by higher priority tasks. In some cases, if the prior tasks have been delayed and next task occurs task collision can occur. Set the priority with due consideration of items such as urgency and execution time of a task.

(3) Task programs are written as shortly as possible?

- If execution time of a task program is long, the scan time may become longer and irregular and also collision of task programs may occur. Therefore, write task programs as shortly as possible.

(4) Protection of lower priority programs against higher priority program isn't needed during execution of those programs.

- If the priority of a task program (or a scan program) has been set to lower priority and other tasks must not interrupt during its execution, use the function DI and 'EI' to protect the program partly. When processing global variables used commonly in other programs, special modules or communications modules, problems can occur.

8) Example of program configuration and processing

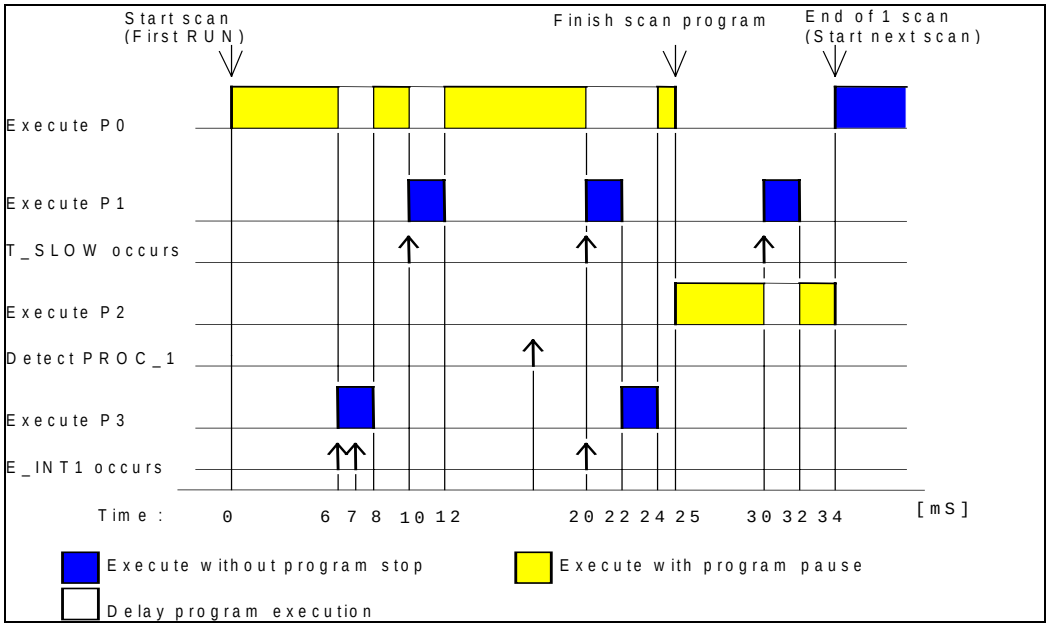
When the task and program have been registered as below,

- Task registration : T_SLOW (interval T#10ms, priority = 0)
  PROC.9 (internal contact point: %MX0, priority = 3)
  E_INT1 (external contact point: %IX0.0.1, priority = 2)
- program registration : program P0
  program P1 with the task T_SLOW
  program P2 with the task PROC_1
  program P3 with the task E_INT1

If program execution time is equal to external interrupt occurrence time:

- Execution time for each program: P0 =17 ms, P1= 2 ms, P2= 7 ms, P3= 2 ms
- Interrupt EINT occurrence time: Occurred at the 6, 7, 20 ms after the operation started.
- PROC_1 occurrence: Invoked during the execution of scan program

Program execution is shown as below.



Processing with time

Time(ms)	Processing
0	Scan starts and the scan program P0 starts its execution.
0~6	The program P0 is being executed.
6~8	Execution request for P3 is input, and P0 is stopped and P3 is executed. Execution request for P1 by E_INT1 at the 7 ms is ignored as the P2 is being executed.
8~10	P3 finishes its execution and the P0 stopped continues its execution.
10~12	P0 is stopped and P1 is executed due to execution request for P1.
12~20	P2 finishes its execution and the P0 stopped continues its execution.
20	Execution requests for P1 and P3 are simultaneously exist, but the higher priority P1 is executed and P3 is ready for its execution.
20~22	P0 is stopped and P1 is executed.
22~24	P1 finishes its execution and the higher priority P3 is executed before P0.
24~25	P3 finishes its execution and the P0 stopped completes its execution.
25	Execution request for P2 is checked at the finish time of the scan program (P0) and P2 is executed.
25~30	The program P2 is executed.
30~32	Execution request for P1 is input and P2 is stopped and P1 finishes its execution.
32~34	P1 finishes its execution and the P2 stopped finishes its execution.
34	A new scan starts. (P0 starts its execution.)

5.4.4 Error Handling

1) Error Classification

Errors occur due to various causes such as PLC system defect, system configuration fault or abnormal operation result. Errors are classified into fatal error mode, which stops system operation for system stability, and ordinary error mode, which continues system operation with informing the user of its error warning.

The main factors that occurs the PLC system error are given as followings.

- PLC hardware defect
- System configuration error
- Operation error during execution of the user programs
- External device malfunction

2) Operation mode at error occurrence

In case of error occurrence, the PLC system write the error contents the corresponding flags and stops or continues its operation complying with its operation mode.

(1) PLC hardware defect

The system enters into the STOP state if a fatal error such as the CPU module defect has occurred, and continues its operation if an ordinary error such as battery error has occurred.

(2) System configuration error

This error occurs when the PLC hardware configuration differs from the configuration defined in the GM7 series. The system enters into the STOP state.

(3) Operation error during execution of the user programs

It the numeric operation error of these errors occurs during execution of the user programs, its contents are marked on the error flags and the system continues its operation. If operation time overruns the watchdog time or I/O modules loaded are not normally controlled, the system enters into the STOP state.

(4) External device malfunction

The PLC user program detects malfunctions of external devices. If a fatal error is detected the system enters into the STOP state, and if an ordinary error is detected the system continues its operation.

REMARK

1) In occurrence of a fatal error the state is to be stored in the representative system error flags, and an ordinary error in the representative system warning flags.

2) For details of flags, refer to Appendix 2. Flag List.

5.4.5 Precautions when using special modules

This system offers convenience and high performance in using special modules compared with the existing methods. Therefore, take some precautions when composing the system. Check the system after the following items have been thoroughly understood.

1) Special module programming

(1) Special function block is offered for each special module to make programs concise and to prevent errors in writing down the user program.

(2) Function block functions as an interface between the user program data and the special modules. As it includes the function that watches the operation status of special modules and indicates the error status, other separate error detection program does not have to be written.

2) Control of special modules

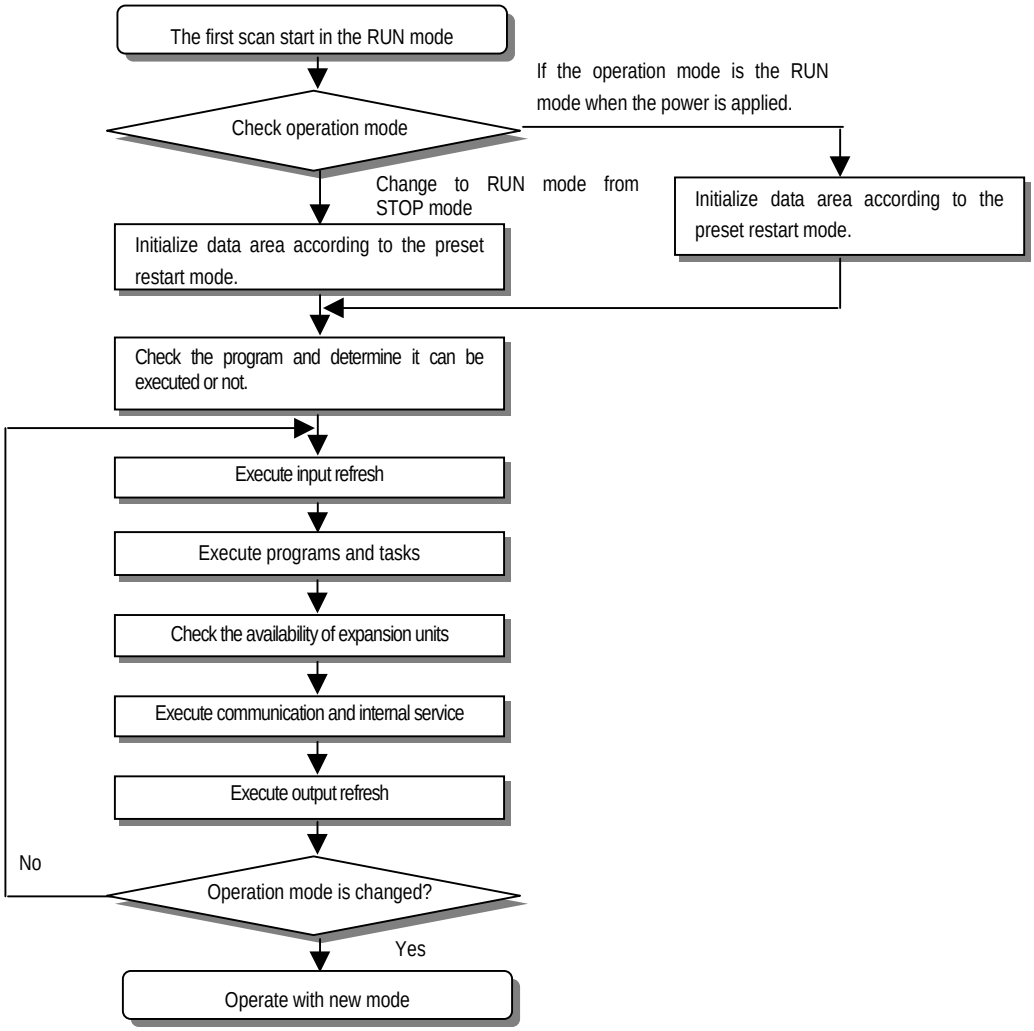
In controlling the operations of special modules, write the program using function blocks, which correspond to the operations that have to be controlled. These function blocks can locate at any place within the program.

5.5 Operation Modes

The CPU module operates in one of the four modes - the RUN, STOP, PAUSE and DEBUG mode. The following describes the PLC operation processing in each operation mode.

5.5.1 RUN mode

In this mode, programs are normally operated.


The first scan start in the RUN mode
If the operation mode is the RUN mode when the power is applied.
Check operation mode
Change to RUN mode from STOP mode
Initialize data area according to the preset restart mode.
Initialize data area according to the preset restart mode.
Check the program and determine it can be executed or not.
Execute input refresh
Execute programs and tasks
Check the availability of expansion units
Execute communication and internal service
Execute output refresh
No
Operation mode is changed?
Yes
Operate with new mode

1) Processing when the operation mode changes.

Initialization of data area is executed when the first scan starts.

(1) If the PLC is in the RUN mode when applying the power:

(2) If the operation mode has been changed into from the STOP mode into the RUN mode : the initialization is executed complying with the restart mode set. (cold / warm / hot)

(3) The possibility of execution of the program is decided with check on its effectiveness.

2) Operation processing contents

I/O refreshes and program operation are executed.

(1) Task programs are executed with the detection of their start-up conditions.

(2) Normal or abnormal operation and mounting conditions of the loaded module are checked.

(3) Communications service or other internal operations are processed.

5.5.2 STOP mode

In this mode, programs are not operated.

1) Processing when the operation mode changes.

The output image area is cleared and output refresh is executed.

2) Operation processing contents

(1) I/O refresh is executed.

(2) Normal or abnormal operation and mounting conditions of the loaded module are checked.

(3) Communications service or other internal operations are processed.

5.5.3 PAUSE mode

In this mode, the program operation is temporarily stopped. If it returns to the RUN mode, the operation continues from the state before the stop.

1) Processing when the operation mode changes

Data area and input image are not cleared and the operating conditions just before the mode change is maintain.

2) Operation processing contents

(1) I/O refresh is executed.

(2) Normal or abnormal operation and mounting conditions of the loaded module are checked.

(3) Communications service or other internal operations are processed.

5.5.4 DEBUG mode

In this mode, errors of a program are searched and the operation sequence is traced. Changing into this mode is only possible in the STOP mode. In this mode, a program can be checked with examination on its execution state and contents of each data.

1) Processing when the operation mode changes

(1) Data area is initialized at the starting time of the mode change complying with the restart mode, which has been set on the parameters.

(2) The output image area is cleared and output refresh is executed.

2) Operation processing contents

(1) I/O refresh is executed by one time every scan.

(2) Communications service or other internal operations are processed.

3) Debug operation conditions

• Two or more of the following four operation conditions can be simultaneously specified.

Operation conditions	Description
Executed by the one (step over)	If an operation command is ordered, the system operates one operation unit operation unit, and stops.
Executed to the specified breakpoint.	• If break step is specified in the program, the operation stops at those step, before execution. .• U to 8 break points can be specified.
Executed according to the contact state	If the contact area to be watched and the condition (Read, Write, Value) where the operation has to stop are specified, the operation stops when the specified operation occurs at the specified contact. (after execution)
Executed by the specified scan number.	If the number of scan that will be operated is specified, the operation stops after it has operated by the specified scan number.

4) Operation method

(1) Execute the operation after the debug operation conditions have been set in the GMWIN.

(2) In task programs, each task can be specified to operation enable/disable.(For detailed operation method, refer to the GMWIN User's Manual Chapter 9.

5.5.5 Operation mode change

1) Operation mode change methods

The following method is used to change the operation mode.

(1) Change by the mode-setting switch of CPU module.

(2) Change by the GMWIN connected with the CPU module communications port.

(3) Change by the GMWIN connected to the remote CPU module through Fnet.

(4) Change by the user's command using FAM or computer link module, etc.

(5) Change by the STOP function', 'ESTOP function' during program execution.

2) Operation mode change by the mode-setting switch of CPU module

The following shows the operation mode change by the mode-setting switch of CPU module.

Mode setting switch position	Operation mode
RUN	Local RUN
STOP	Local STOP
STOP → PAU / REM	Remote STOP
PAU / REM → RUN ∗ 1	Local RUN
RUN → PAU / REM * 2	Local PAUSE
PAU / REM → STOP	Local STOP

REMARK

1) ∗ 1: If the operation mode changes from RUN mode to local RUN mode by the mode setting switch, the PLC operates continuously without stopping.

2) * 2 : If Local PAUSE disable (or Local PAUSE enable) is set by parameter in GMWIN, it operated as Remote RUN (or Local PAUSE).

3) Remote operation mode change

Remote operation mode change is available only when the operation mode is set to the remote STOP mode (i.e., the mode setting switch position is in the STOP PAU/REM').

Mode setting switch position	Mode Change	Mode change by the GMWIN	Mode change using FAM or computer link, etc.
PAU / REM	Remote STOP → Remote RUN		
	Remote STOP → Remote PAUSE	X	X
	Remote STOP → DEBUG		
	Remote RUN → Remote PAUSE		
	Remote RUN → Remote STOP		
	Remote RUN → DEBUG	X	X
	Remote PAUSE → Remote RUN		
	Remote PAUSE → Remote STOP		
	Remote PAUSE → Remote DEBUG	X	X
	DEBUG → Remote STOP		
	DEBUG → Remote RUN	X	X
	DEBUG → Remote PAUSE	X	X

4) Remote operation mode change enable/disable

It is possible to disable the mode change for system protection so that some parts of the operation mode sources cannot change the mode. If remote operation mode change has been disabled, the operation mode change is possible only by the mode setting switch and GMWIN. To enable the remote operation change, set the parameter 'Enabling the PLC control by communications' to enable. (For details, refer to the Appendix 1. System Definitions)

5.6 Functions

5.6.1 Restart mode

The restart mode defines how to initialize variables and the system and how to operate in the RUN mode when the system starts its operation with the RUN mode by re-application of the power or mode change. Two restart modes, cold and warm restart are available and the execution condition for each restart mode is given below.

(For details, refer to the 4.5.1 Basic Parameters Edit' of the GMWIN User's Manual Section 4.5 Parameters Edit.

1) Cold Restart

(1) It is executed when the restart mode parameter has been set to the cold restart mode.

(2) All data are cleared as '0' and only variables of which initial value has been defined will be set as their initial value.

(3) Though the parameter has been set to the warm restart mode, cold restart will be executed at the first execution of a program after it has been changed.

(4) In case of selection 'Reset' command in the GMWIN, it restarts in accordance with setting in parameter and in case of selection 'Overall Reset' command, it restarts as cold restart mode.

2) Warm Restart

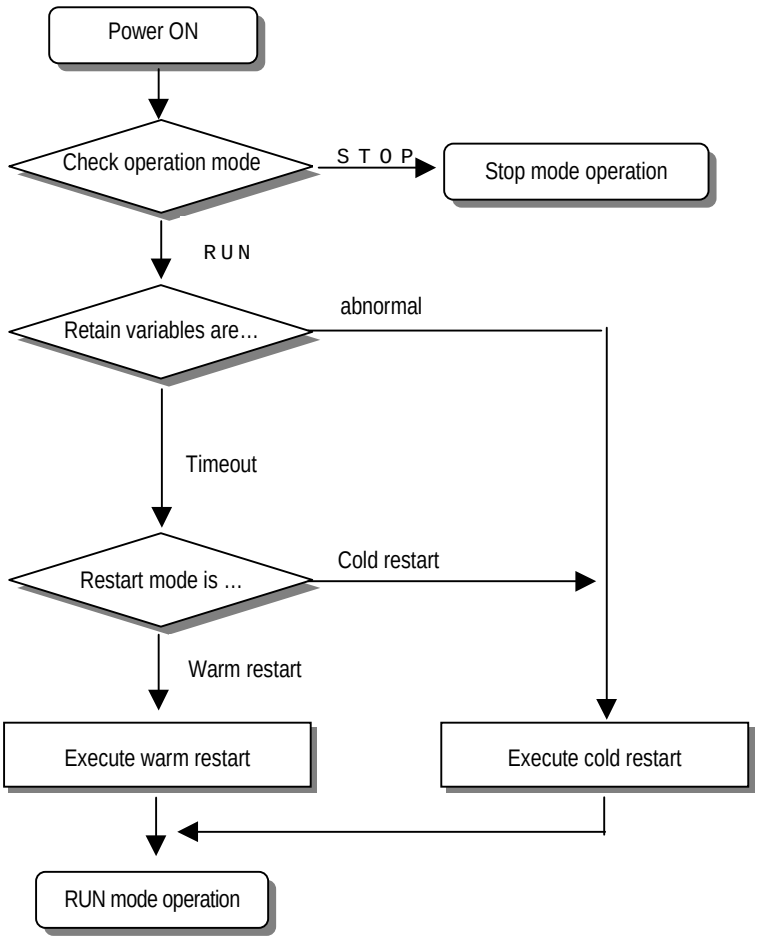
(1) It is executed when the restart mode parameter has been set to the warm restart mode.

(2) A data which set as retain & initial will be retain and a data which set as initial value will be set with default value during the warm restart. All other data will be cleared with '0'.

(3) Though the parameter has been set to the warm restart mode, cold restart will be executed at the first execution of a program after it has been stopped due to its down load and error.

(4) Though the parameter has been set to the warm restart mode, cold restart will be executed if data contents are abnormal (i.e., the data does not remain at a power failure)

- Restart mode is executed as the figure given below when the power has been re-applied during execution of the CPU module



4) Data initialization according to the restart mode

The variables relating to the restart mode are classified into three types, i.e, default variable, initialization variable and retain variable. The following table shows the initialization method for each type variable.

Mode / Variable type	COLD	WARM
Default	Initialized with 0’	Initialized with 0’
Retain	Initialized with ‘0’	Previous value is retained.
Initialization	Initialized with the user defined value	Initialized with the user defined value
Retain & Initialization	Initialized with the user defined value	Previous value is retained.

REMARK

Definitions of variable

(1) Default variable: A variable whose initial value is not defined or previous value will not be retained.

(2) Initialization variable: A variable whose initial value is defined.

(3) Retain variable: A variable whose previous value will be retained.

5.6.2 Self-diagnosis

1) Functions

(1) The self-diagnosis function permits the CPU module to detect its own errors.

(2) Self-diagnosis is carried out when the PLC power supply is turned on and when an error occurs the PLC is in the RUN state. If an error is detected, the system stops operation to prevent faulty PLC operation.

2) Error flag

If an error occurs, it will be stored to the following flags and the STOP LED flickers.

- Representative system error flag: _CNT_ER
- Representative system warning flag: _CNF_WAR

REMARK

1) Refer to 11.5 'Error Code List of Chapter 11'. Troubleshooting for details of contents of self-diagnosis and corrective actions.

5.6.3 Remote function

The CPU module can be controlled by external operations (from GMWIN and computer link module, etc.). For remote operation, set the mode setting switch of CPU module to remote position.

1) Remote RUNISTOP

(1) The remote RUN/STOP permits external operations to RUN/STOP the CPU module under the condition that the mode selling switch of CPU module is in the remote position.

(2) This function is convenient when the CPU module is located on the place where it is difficult to control the CPU module or the user want to control the CPU module in the control panel from outside.

2) Remote PAUSE

(1) The remote PAUSE permits external operations to execute PAUSE operations under the condition that the mode-setting switch of CPU module is in the remote position. The PAUSE operations stop the CPU module operation processing while maintaining the On/Off state of the output module.

(2) This function is convenient when the user wants to maintain the ON state of the output module under the condition the CPU module has been stopped.

3) Remote DEBUG

(1) This function permits external operations to execute DEBUG operations under the condition that the mode setting switch of CPU module is in the remote position. The DEBUG operations execute programs complying with the specified operation conditions.

(2) This function is convenient when program execution or contents of any data are checked for debugging of the program.

4) Remote reset

(1) This function permits remote operations to reset the CPU module, which locates in the place where direct operations cannot be applied, when an error has occurred.

REMARK

1)For remote function operations, refer to the GMWIN User's Manual Chapter 7. On-line.

5.6.4 I/O Force On/Off function

1) Force On/Off setting method.

Force on/off setting is applied to input area and output area.

Force on/off should be set for each input and output, the setting operates from the time that Force I/O setting enable' is set.. This setting can be done when I/O modules are not really loaded.

2) Force on off Processing timing and method

(1) Force Input

- After data have been read from input modules, at the time of input refresh the data of the junctions which have been set to force on/off will be replaced with force setting data to change the input image area. And then, the user program will be executed with real input data and force setting data.

(2) Force output

- When a user program has finished its execution the output image area has the operation results. At the time of output refresh the data of the junctions which have been set to force on/off will be replaced with force setting data and the replaced data will be output. However, the force on/off setting does not change the output image area data while it changes the input image area data.

(3) Force on off processing area

- Input/output areas for force on/off setting are larger than the real I/O areas. If remote I/O is specified using this area, the force on/off function is as just available in it as in the basic I/O areas.

(4) Precautions

- Turning the power off and on, changes of the operation mode or operation by reset switch (GM3) does not change the previous force on/off setting data. They remain within the CPU module and operation is executed with the same data.
- Force I/O data will not be cleared even in the STOP mode.
- If a program is downloaded or its backup breaks, the force on/off setting data will be cleared. The operating program in memory differs from the program in the flash memory so that if operation restarts with the program in the flash memory the on/off setting data will be also cleared.
- When setting new data, disable every I/O settings using the setting data clear' function and set the new data.

REMARK

1) For detailed operation, refer to the GMWIN user's Manual Chapter 7 'Force I/O setting.

5.6.5 Direct I/O Operation function

This function is usefully available when an input junction state is directly read during execution of a program and used in the operation, or the operation result is directly output to an output junction.

1) Direct input

- direct input is executed by use of the 'DIRECT_IN' function. If this function is used, the input image area will be directly updated and applied to the continuing operations.

2) Direct output

- Direct output is executed by use of the 'DIRECT_O' function. If this function is used, the data of the output image area, which has the operation results by the time, will be directly output to the direct output module.

3) Force on/off

- Force on/off settings are still effective when processing direct I/O.

5.6.6 External device error diagnosis function

Flags are given for the user to implement easily the program in which the error detection of external devices and system stop and warning are coded. By use of these flags, error indication of external devices is possible without complex programming and monitoring of the error location can be done without special tools (GMWIN, etc.) or source programs.

1) External device fault detection and classification

(1) The user program detects external device faults. The faults are classified into fatal fault (error), where the PLC stops its operation, and ordinary fault (warning), where operation continues.

(2) The flag ANC_ERR[n] is used to indicate error. The flag ANC_WN[n] is used to indicate warning.

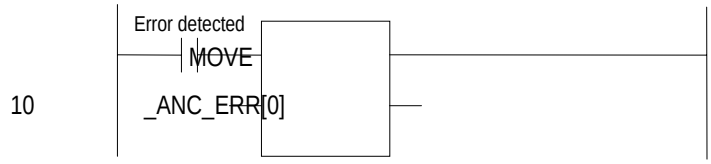
2) External device fatal-fault (error) processing

(1) If an error of external device is detected and the error type, where other value than 0 is used, is written to the system flag ANC_ERR[n], the flag will checked at the time that scan program finishes its execution. If an error is indicated on the flag, it will be also indicated on the _ANNUN_ER of the representative system error flag _CNF_ER, the PLC turns all output modules off and the error state will be same as the PLC self-diagnosis.

(2) The user can know the cause of error by use of the GMWIN, and also by direct monitoring of the flag _ANC_ERR[n].

(3) As the flag _ANC_ERR[n] has sixteen elements (n: 0 to 15), the user can classify error states largely. User defined error No. can be written to the elements. A number of 1 to 65535 is usable.

■ Example)



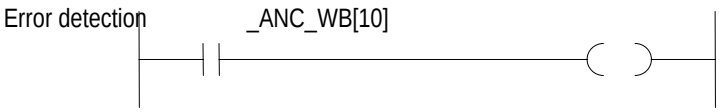
3) External device Ordinary-fault (Warning) Processing

(1) If a warning of external device is detected and the corresponding flag of the system flag _ANC_WB[n] is set to on, the flag will checked from the _ANC_WB[0] at the time that scan program finishes its execution. If an error is indicated on the flag, it will be also indicated on the _ANNUN_WR of the representative system warning flag _CNF_WAR. External device waning numbers will be written to from _ANC_WAR [0] to ANC.WAR [7] according to occurrence sequence.

(2) The user can know the cause of error by use of the GMWIN, and also by direct monitoring of the flags _ANC_WAR[n] and _ANC_WB[n].

(3) If an external device waning is removed, that is, the elements of _ANC_WAR[n] are released from warning, the corresponding _ANC_WAR[n] will be automatically cleared, If all element flags are cleared, the flag _ANNUN_WR of the system flag _CNF_WAR will be reset.

■ Example)



_ANNUN_WR = 1
_ANC_WAR[0] = 10
_ANC_WAR[1] = 0
_ANC_WAR[2] = 0
_ANC_WAR[3] = 0
_ANC_WAR[4] = 0
_ANC_WAR[5] = 0
_ANC_WAR[6] = 0
_ANC_WAR[7] = 0

If the user program had detected a system fault and set _ANCWB[10] to ON, the states of _ANNUNWR and _ANN WAR [0.7] will be shown as left after the scan has been finished.

_ANNUN_WR = 1
_ANC_WAR[0] = 10
_ANC_WAR[1] = 1
_ANC_WAR[2] = 2
_ANC_WAR[3] = 3
_ANC_WAR[4] = 15
_ANC_WAR[5] = 40
_ANC_WAR[6] = 50
_ANC_WAR[7] = 60

After the next scan has been finished, if the numbers 1, 2, 3,10,15 40, 50, 60 and 75 of _ANC_WB[n] are tuned on _ANC_WAR[n] will be shown as left.

As the number 10 has turned on (has occurred) in the previous scan, though the number 10 has lower priority than the numbers 1, 2 and 3, it will be the lower element of _ANCWAR[n]. The _ANC_WB[75] is not indicated as it is turned on and the warning that occurred before has written to the _ANCWARIn1.

_ANNUN_WR = 1
_ANC_WAR[0] = 1
_ANC_WAR[1] = 2
_ANC_WAR[2] = 3
_ANC_WAR[3] = 15
_ANC_WAR[4] = 40
_ANC_WAR[5] = 50
_ANC_WAR[6] = 60
_ANC_WAR[7] = 75

After the next scan has been finished, if the numbers 1, 2, 3,10,15 40 , 50, 60 and 75 of _ANC_WB[n] are tuned on _ANC_WAR[n] will be shown as left.

The No. 10 warning has been released the content of _ANCWAR[0] will be cleared and the contents of _ANC_WAR[1..7] will shift into the lower elements. The content of _AN&WAR[7] will has been cleared by the shifting and the content of _AN&WB[751 will be written to ANCWAR[7].

_ANNUN_WR = 0
_ANC_WAR[0] = 0
_ANC_WAR[1] = 0
_ANC_WAR[2] = 0
_ANC_WAR[3] = 0
_ANC_WAR[4] = 0
_ANC_WAR[5] = 0
_ANC_WAR[6] = 0
_ANC_WAR[7] = 0

If all warnings indicated on the _ANC_WB[n] are released during operation, the ANNUN_WR and _ANC_WAR[n] will be shown as left.

5.7 Memory Configuration

The CPU module includes two types of memory that are available by the user. One is program memory, which is used to store the user programs written to implement a system by the user. The other is data memory, which stores data during operation.

1) Program memory configuration

The table given below shows the contents to be stored and the storage capacity of program memory.

Item	Memory Capacity
Overall program memory area	68 k byte
Parameter area • Basic parameter area • High speed link parameter area • interrupt setting information area	2.0 Kbytes
Program area • Scan program area • Task program area • User defined function/function block area • Standard library area • Variable initialization information area • Protective variable specification information area	66 Kbytes

2) Data memory Configuration

The table given below shows the contents to be stored and the storage capacity of program memory.

Item	Memory Capacity
Overall data memory area	32 Kbytes
System area • I/O information table • Force I/O table	1 Kbytes
System flag area	1.5 Kbytes
Input image area (%IX)	128 byte
Output image area %QX	128 byte
Direct variable area %M	2 ~ 8 Kbytes
Symbolic variable area (maximum)	29 Kbytes – the size of direct variable area

3) Purpose

(1) System area

It used to save the self-producing data of the CPU module for the system management and GMWIN system control data.

(2) System flag area

It used to save the user flags and system flags. The user operates it by flag names.

(3) Input image area

It used to save input data read from input modules. Overall size is %IX0.0.0~%IX0.7.63. Only %QX0.0.0~%QX0.3.63 can be used as a real input domain but the other unused domain can be used as convenience, especially remote output data for communication can be saved here as convenience.

(4) Output image area

It used to save operation results that are automatically output through the output device. Overall data size is %QX0.9.9~%QX0.7.63. In GM7, only %QX0.0.0~%QX0.3.63 can be used as a real input domain but the other unused domain can be used as convenience, especially remote output data for communication can be saved here as convenience.

(5) Direct variance area

The user can use this area to access direct memory data, using the variable names such as %MX0, %MB0, %MW0, %MD0, which was defined in advance by the system. Memory size is defined when the user makes program. Refers to “system definitions” for the variable area available to use according to the setting.

(6) Symbolic variable area

It used to save the variables that when the user creates a program or when the user defines a global variables, is automatically allocated its memory. The variables used in program blocks are located in the ‘PB instance memory’ of the related program, and the memory used in the function block is located in the ‘FB instance memory.’

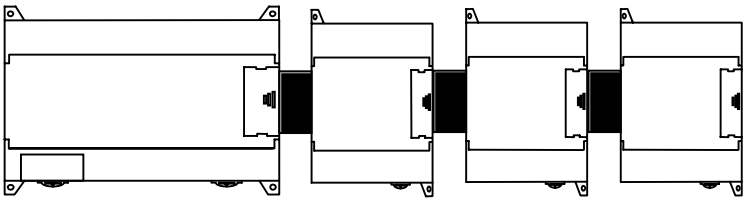
5.8 I/O No. Allocation Method

1) I/O No. allocation means to give an address to each module in order to read data from input modules and output data to output modules.

2) Fixed 64 points are allocated to each module for I/O points.

The following shows an example of I/O No. allocation method.

	Base unit(20~60 points)	Expansion module(10 points)	Expansion module(AD mix)	Expansion module(10 points)
Input	%IX0.0.0 ~ %IX0.0.35	%IX0.1.0 ~ %IX0.1.5	-	%IX0.3.0 ~ %IX0.3.5
Output	%QX0.0.0 ~ %QX0.0.23	%QX0.1.0 ~ %QX0.1.3	-	%QX0.3.0 ~%QX0.3.3

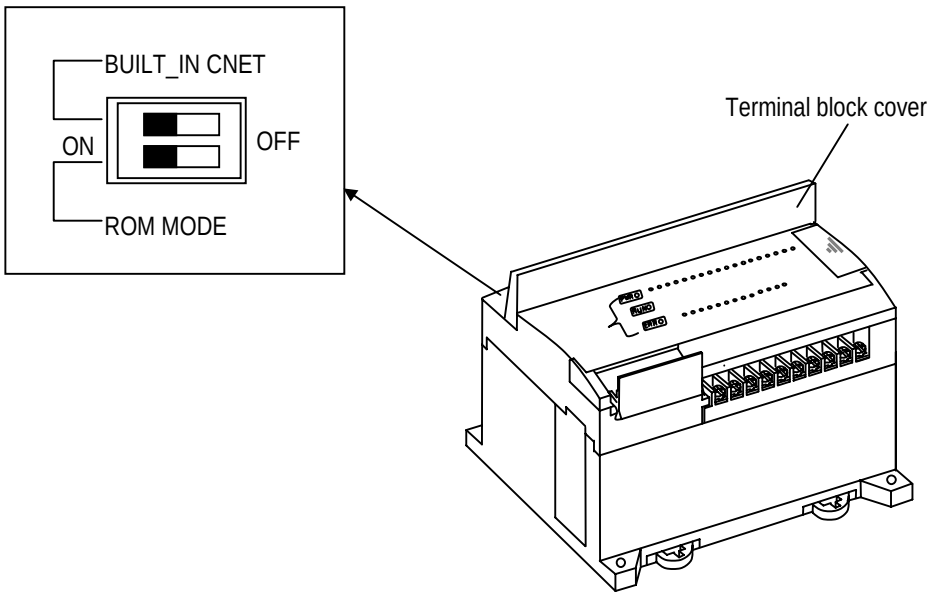


5.9 Built-in Flash Memory

GM7 series includes a built-in flash memory to store user program. Also, user can set the PLC automatically executes the user program of flash memory when the PLC is turned on. It is similar with the ROM operation of other PLCs, but it is different that no external memory is required.

5.9.1 Structure

You can see dip switches as shown when you open I/O terminal block cover.

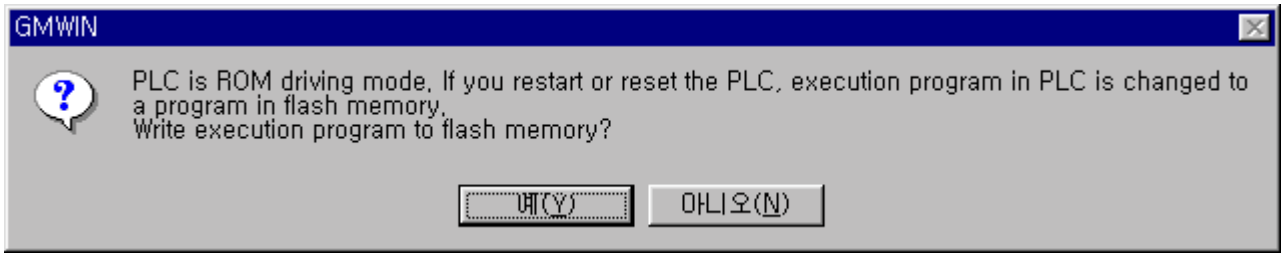


5.9.2 Usage

1) Set the base unit to the STOP mode.

2) There are 2 ways to use writing program into the built-in flash memory of base unit.

(1) While ROM_MODE switch is On, if the program is written on the basic unit, the following message at GMWIN shows.


GMWIN
PLC is ROM driving mode. If you restart or reset the PLC, execution program in PLC is changed to a program in flash memory.
Write execution program to flash memory?
예(Y)
아니오(N)

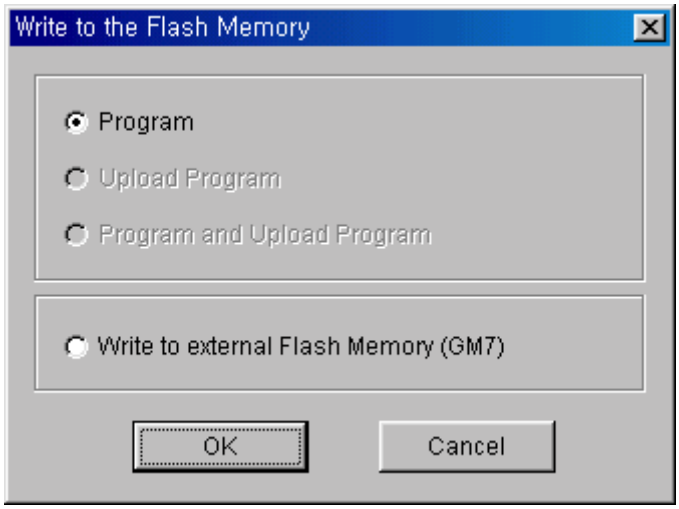
∅ If you want to input downloaded program into flash memory, press 'y.'

REMARK

Above message shows only when ***Online – Writing – Parameter and program*** is chosen in menu.

(2) To save in the flash memory, lest not write on the program, it's as follows:

∅ If ***Online – Flash memory – Writing on flash memory*** is chosen, the following window shows.



∅ Choose title to be saved in the flash memory and press 'y.'

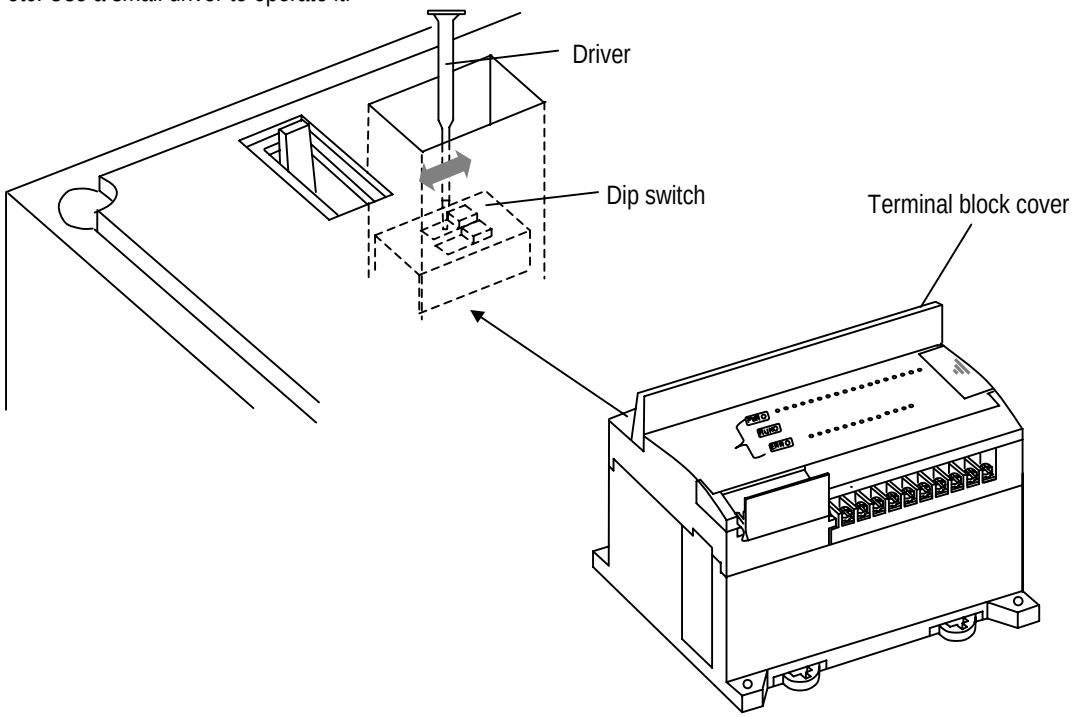
3) When you reset the PLC system, it works according to the dip switch for operating flash memory.

Dip switch position	Description
upper switch is for Cnet. ON OFF ROM MODE	When power is on, the program saved in the flash memory operates.
Upper switch is for Cnet. ON OFF ROM MODE	CPU recognizes that there is no program in the flash memory, and starts to drive program from RAM.

REMARKS

1) The flash memory dipswitch has no relationship with saving the program.
When you save the program in the flash memory, set STOP mode as the operation mode of the basic units. If you try to save when the mode is RUN, the following window shows.

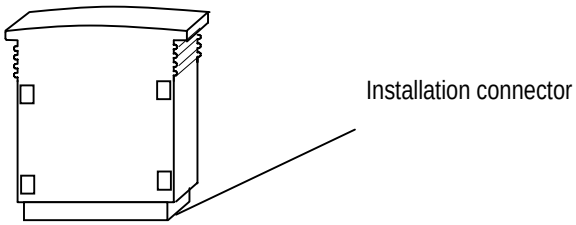
3) Dip switch for flash memory operation is placed in deep place to prevent a mistaken operation caused by terminal block cover, etc. Use a small driver to operate it.



5.10 External Memory Module

GM7 series supplies external memory module for the user to save programs safely or download a program on the system and use it in case of a program is damaged.

5.10.1 Structure



5.10.2 Usage

1) Saving the user's program on the external memory module.

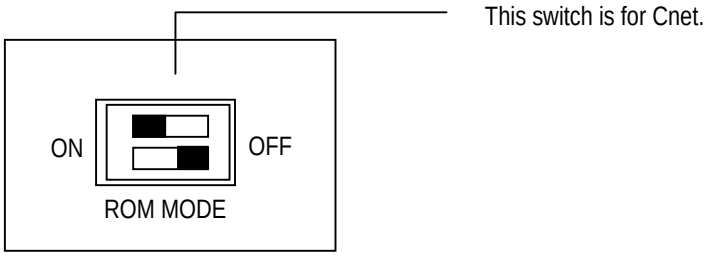
Turn the power of the base unit off.

Install the memory module.

∅ When only basic unit is used: Connect to the expansion connector of the basic unit.

∅ When expansion unit is used: Connect to the expansion connector of the last connected expansion unit.

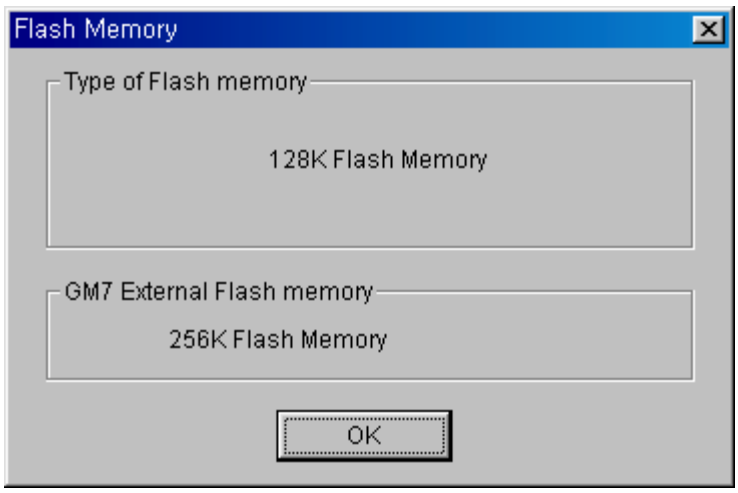
Turn the dip switch for ROM mode setting of the base unit to OFF.



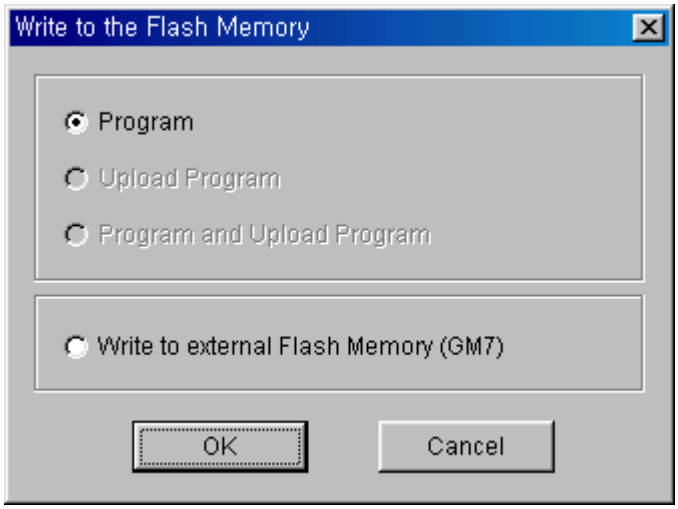
(4) Turn the power of the base unit on.

(5) Connect GMWIN and PLC.

(6) Select ***Online – Flash memory – Type information*** to confirm the flash memory size and installation of the memory module.



(7) Choose ***Online – Flash memory – Writing*** in menu, and the following message box will displayed.



(8) Choose an item to be saved in the flash memory and press 'OK.'

(9) Turn the power of the base unit off.

(10) Remove the external memory module.

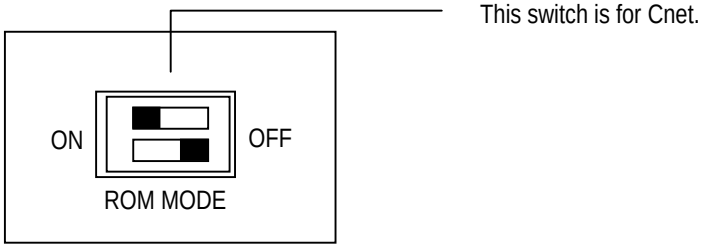
Through the above steps a user can save a program into the external memory module.

2) Run the PLC with a program of external memory module

(1) Turn the power of the base unit off.

(2) Install the memory module (When only base unit is used, connect to the expansion connector of the base unit. And when expansion unit is used, connect to the expansion connector of the last connected expansion unit).

(3) Set the dip switch for ROM mode setting of the base unit to OFF position.



(4) Turn on the power of the base unit.

(5) As RUN LED and ERR. LED are on, the contents of the memory module is transferred into the program area of the base unit and ROM operation area of the flash memory. (It may take about 15 sec.)

(6) Operate according to the set operation mode.

(7) Turn off the power of the basic unit.

(8) Remove the memory module.

(9) Turn the power on.

Through the above steps the user can operate the PLC with program stored in the external memory module.

REMARK

1) When the PLC is operated with the external memory module, it always operates with cold restart.

2) Remove after the program transfer is finished.

5.11 Battery

1) Specifications

Item	Specifications
Normal voltage	DC 3.0 V
Warranty life time	5 years
Application	Programs and data backup, and RTC runs in power failure
Specifications	Lithium Battery, 3V
External dimension (mm)	ϕ 14.5 X 26

2) Handling Instructions

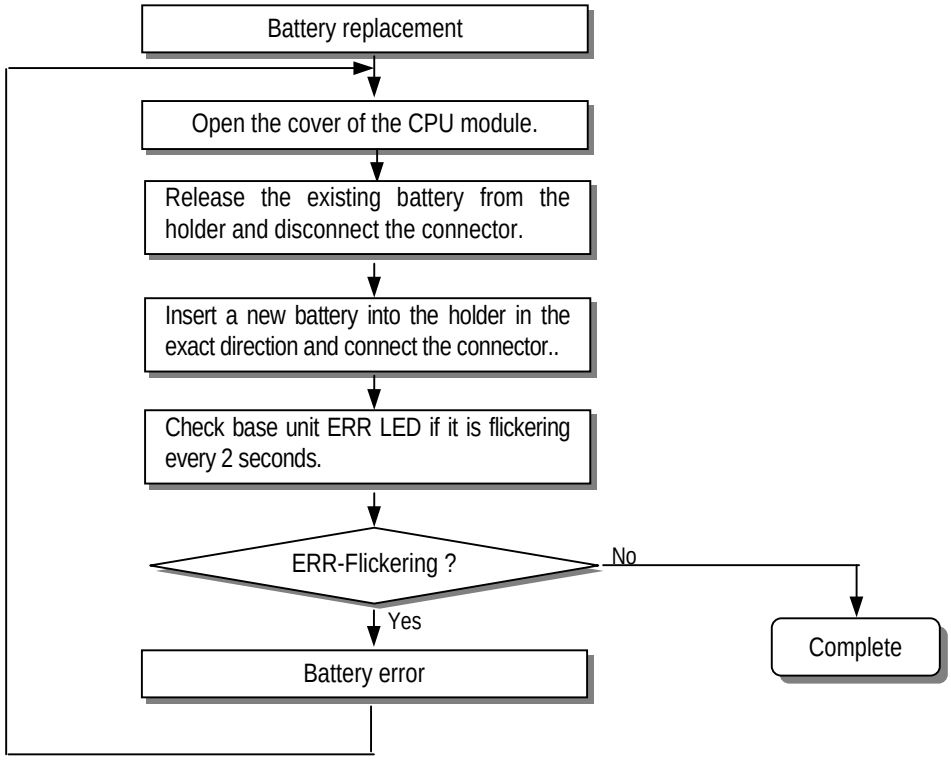
(1) Don't heat or solder its terminals.

(2) Don't measure its voltage with a tester or short circuit.

(3) Don't disassemble.

3) Battery Replacement

Backup battery needs periodic exchange. In case of battery replacement at power off, the built-in super capacitor backup the program and retain variables about 30 minutes. However, it is recommended to complete the battery replacement as soon as possible, or turn on the base unit during battery replacement.


Battery replacement
Open the cover of the CPU module.
Release the existing battery from the holder and disconnect the connector.
Insert a new battery into the holder in the exact direction and connect the connector..
Check base unit ERR LED if it is flickering every 2 seconds.
ERR-Flickering ?
No
Yes
Complete
Battery error

5.12 RTC module

The G7E-RTCA module GM7 provides RTC(Real Time Clock) function for GM7 series. The RTC function can be used for time-scheduling control or recording an error occurrence time. The RTC data is updated into system operation status flag per every scan.

1)Functions

The RTC module will send the RTC data to main unit per every scan.

By the super capacitor back-up, the RTC module keep operating while the power is off.

2)RTC data

Item	Data
Year	Upper 2 digit of year data
	Lower 2 digit of year data
Month	1 ~ 12
Date	1 ~ 31
Hour	0 ~ 23 (24 hour)
Minute	0 ~ 59
Second	0 ~ 59
Day	0 ~ 6 (Monday:0 ~ Sunday:6)
Century	Indicate upper 2 digit of year data

(1)Accuracy

Max. ±2.2 sec / 1 day (At 25 °C)

(2)RTC data back-up time

200 hours (At 25 °C)

(3)Read/Write of RTC data

Select the menu 'Online – PLC Information' win GNWIN software.

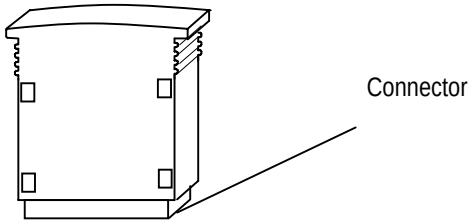
NOTE

1) The RTC module is sold with no initial RTC data setting. Be sure to input the RTC data when use a RTC module first time.

2) The RTC module may show abnormal operation when a improper RTC data is written.
Example) 14(Month) 32(Date) 25(Hour)
When the super capacitor is fully discharged, the RTC module may stop operation or RTC data may be broken. To remove error status, re-write RTC data by GMWIN software.
The system flag _RTC_ERR of _CNF_WAR will turn on when a RTC data error occurred. This flag will turn off automatically when the error is cleared.

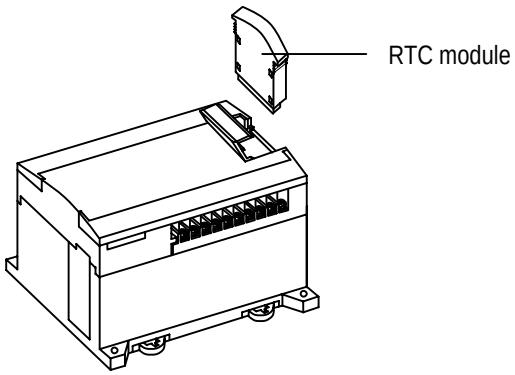
3) Shape of module and connection method

(1) Shape of module

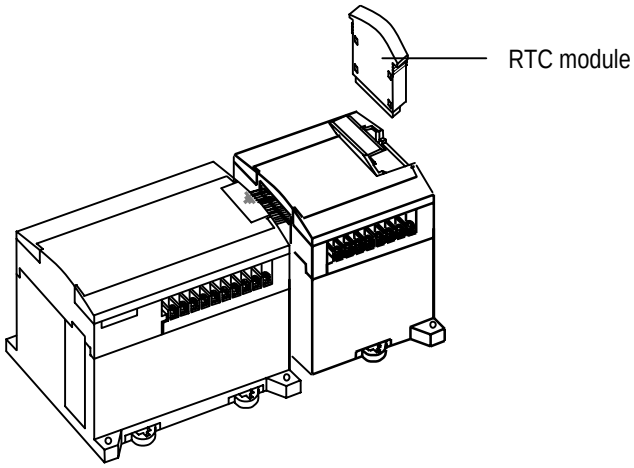


(2) Connection method
Insert the RTC module into the expansion connector of the base/expansion module at last position.

When use base unit only



When use base unit and expansion unit

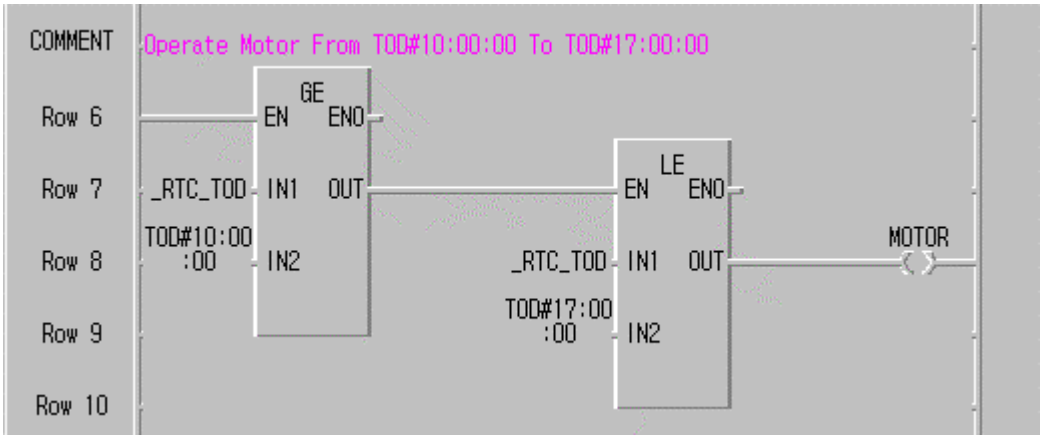


4) Read RTC data

Example : 1998. 12. 22. 19:37:46, Tuesday

Keyword	Type	Name	Description	Data
_RTC_TOD	TOD	RTC present time	Time Data	TOD#19:37:46
_RTC_WEEK	UINT	RTC present day	Day data *(0: Monday, 1:Thuesday, 2: Wednesday, 3: Thursday, 4: Friday, 5: Saturday, 6:Sunday)	1
_INT_DATE	DATE	RTC present date	Date Data of standard format (Reference date – Jan. 1, 1984)	D#1998-12-22
_RTC_ERR	BOOL	RTC data error	This flag indicates that RTC DATA error.	0
_RTC_TIME[n] * n : 0 to 7	BCD	Present time	BCD data of present time of RTC _RTC _TIME[0] : year, _RTC _TIME[1] : month, _RTC _TIME[2] : day, _RTC _TIME[3] : hour, _RTC _TIME[4] : minute, _RTC _TIME[5] : second, _RTC _TIME[6] : day of the week, _RTC _TIME[7] : century Day of the week : 0 : Mon., 1: Tue., 2: Wed., 3:Thur., 4:Fri., 5: Sat., 6:Sun.	_RTC _TIME[0] : 16#98 _RTC _TIME[1]: 16#12 _RTC _TIME[2] : 16#22 _RTC _TIME[3] : 16#19 _RTC _TIME[4] : 16#37 _RTC _TIME[5] : 16#46 _RTC _TIME[6] : 16#1 _RTC _TIME[7] : 16#19

Example Program

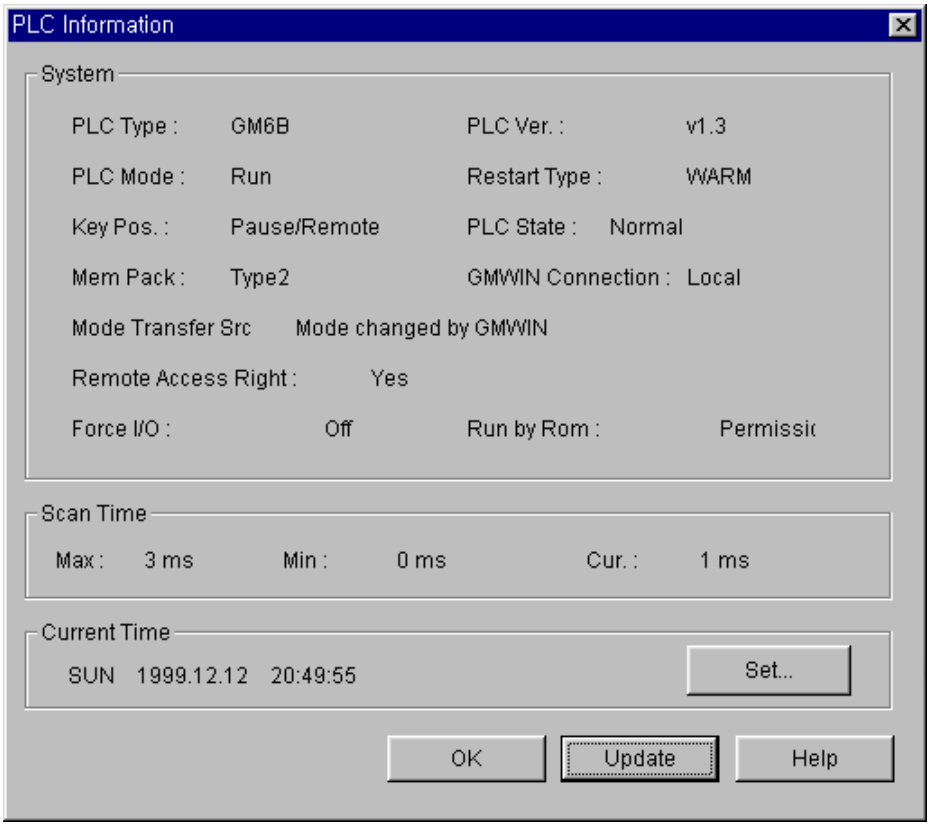


5) Write RTC data

There is two ways to write new RTC data to the CPU.

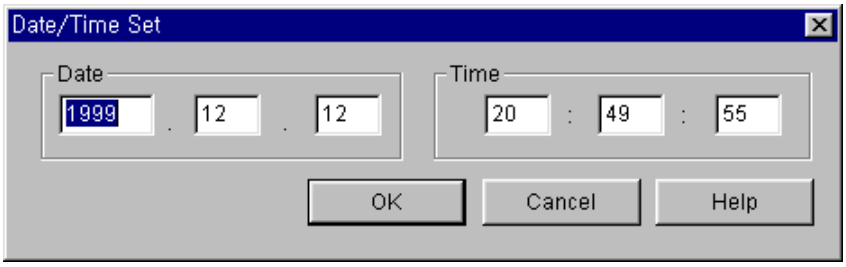
The first one is using GMWIN.

◆ Select **Online-PLC Information-System Info..** in the pull-down menu.



If you want to setup or edit current time,

◆ Select **Set…** button of current time in **PLC Information** dialog box.



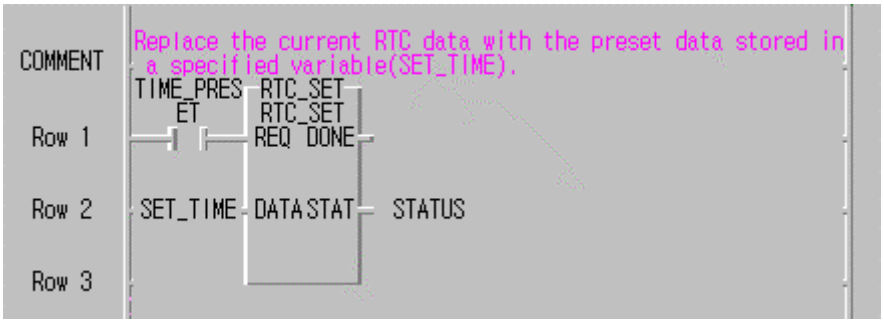
◆ Setup Date and Time in **Date-Time Set** dialog box.

For detailed information, refer the user's manual of GMWIN.

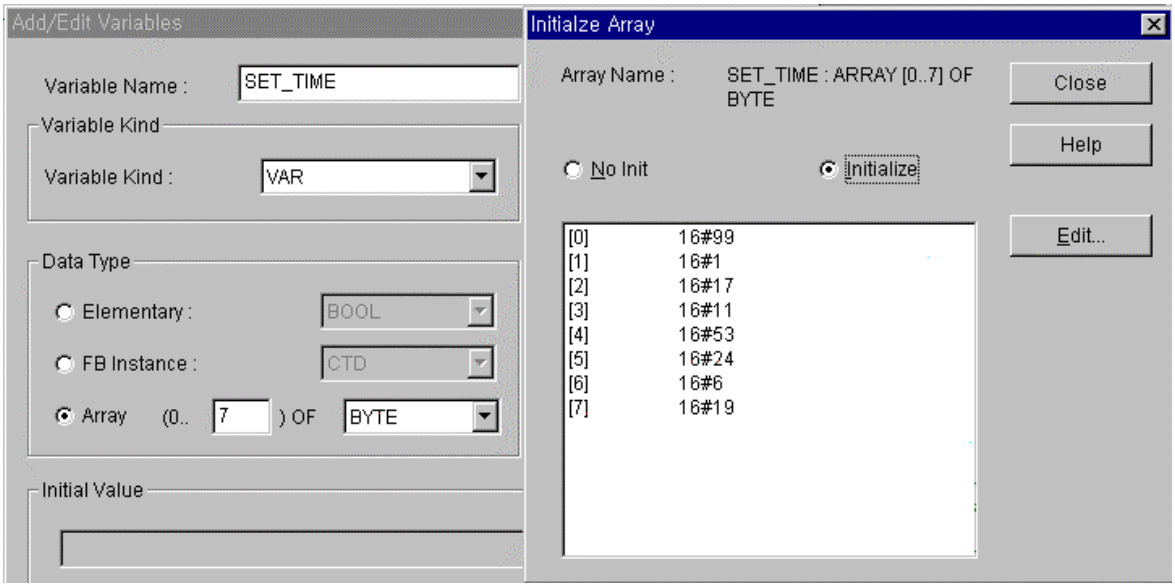
The second one is write sequence program. By executing a F/B(RTC_SET), user can replace the current RTC data with the preset data stored in a specified variable. The following is an example program.

Example The preset RTC data*: 1999. 1. 17. 11:53:24, Sunday*

When the 'TIME_PRESET' bit is switched on, the new data in 'SET_TIME' will be moved to '_RTC_TIME'.



*SET_TIME Variable Setting



Error code of F/B

The following table shows error codes appear at the STAT output.

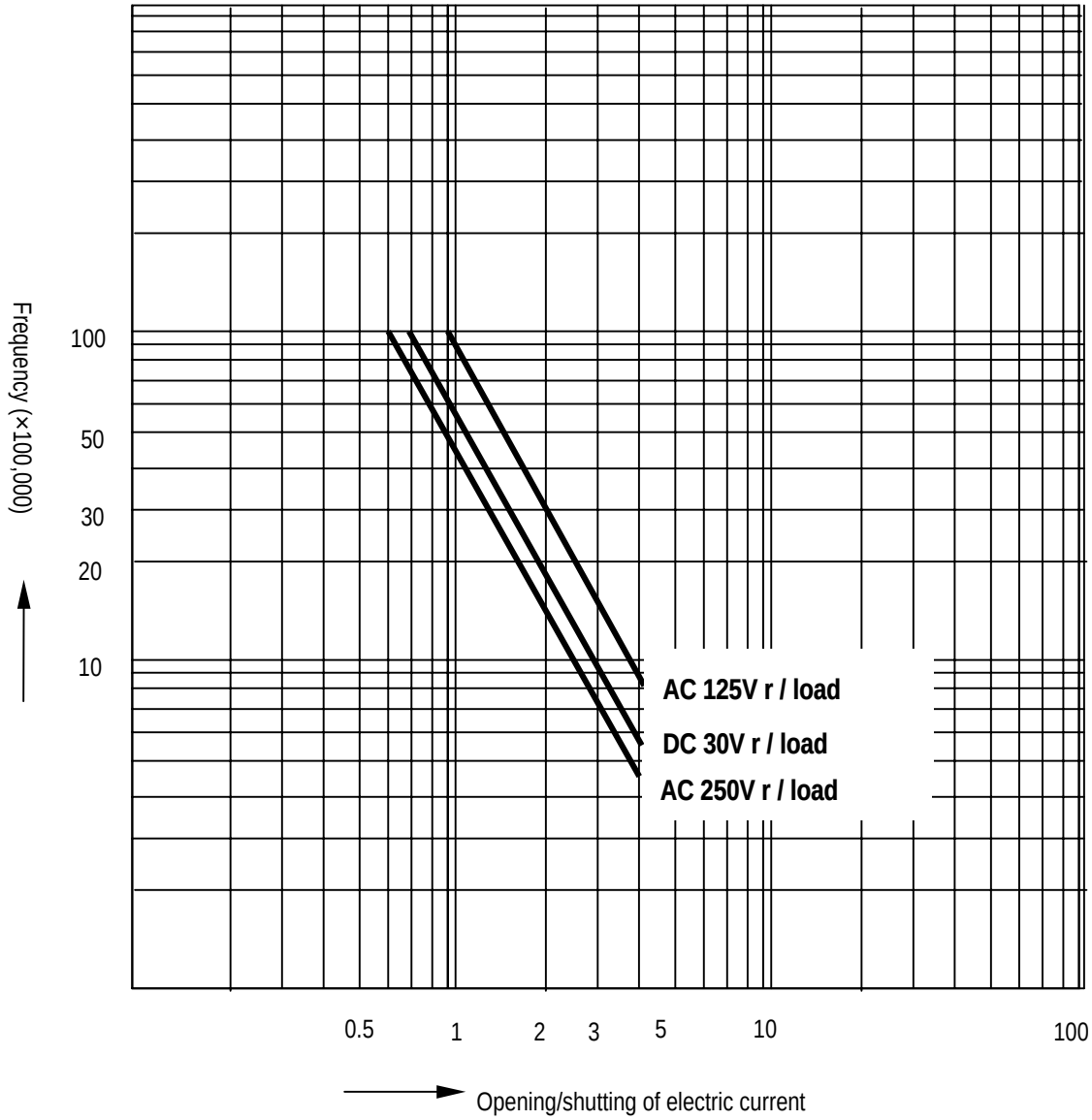
Error code	Description
00	No error
01	RTC Module is not found * Insert the RTC module into the expansion connector
02	A improper RTC data is written. Ex) 14(Month) 32(Date) 25(Hour) * Please write a correct RTC data

Chapter 6 Input and Output Modules

6.1 Input / Output Specifications

Digital input that offers to IMO–GM7 series are made to use both of electric current sink and electric current source. To keep use coil load as an output module, maximum opening and shutting frequency is 1 second on and 1 second off.

The following diagram shows maximum life relay for relay output.



6.2 Digital Input Specification

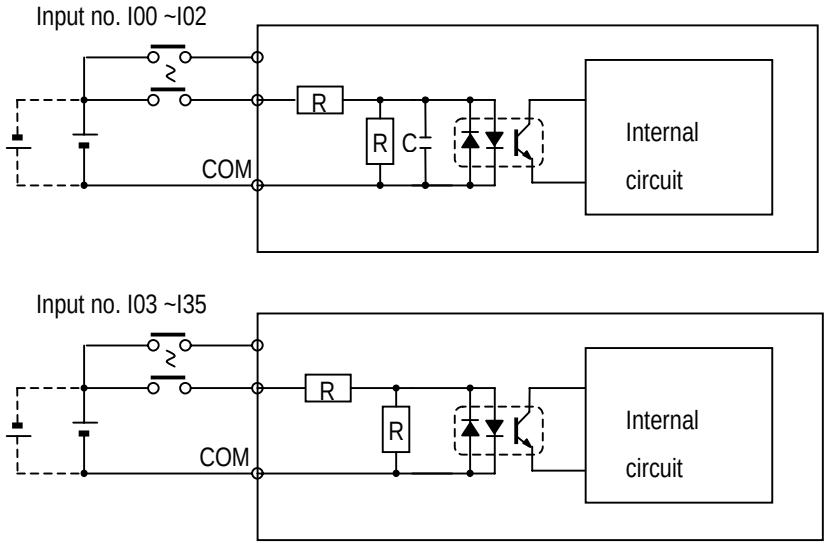
6.2.1 Base unit

1) Specification

Model / Specification		Base unit				
		G7M-DR10A G7M-DR10A/DC G7M-DT10A	G7M-DR20A G7M-DR20A/DC G7M-DT20A	G7M-DR30A G7M-DR30A/DC G7M-DT30A	G7M-DR40A G7M-DR40A/DC G7M-DT40A	G7M-DR60A G7M-DR60A/DC G7M-DT60A
Number of input points		6 points	12 points	18 points	24 points	36 points
Insulation method		Photo coupler				
Rated input voltage		DC 12 / 24V				
Rated input current		4.5 / 9 mA (I00 ~ I02 : 8 / 16mA)				
Operating voltage range		DC10.2 ~ 28.8V (ripple: less than 5%)				
Max. simultaneous input points		100% simultaneously On				
On voltage / On current		DC9.5V or higher/ 4.3 mA or higher (I00 ~ I02 : 6.3mA or higher)				
Off voltage / Off current		DC5V or lower / 1.8 mA or lower (I00 ~ I02 : 3.3mA or lower)				
Input impedance		Approx. 2.7 kΩ (I00~I02: approx. 1.5 kΩ)				
Response time	Off → On	15ms or less * 1				
	On → Off	15ms or less * 1				
Common terminal		12 points / COM	18 points / COM	12 points / COM	18 points / COM	
Operating indicator		LED turns on at ON state of input				

* 1: It is possible to select from 1ms to 15ms by 1ms at GMWIN.

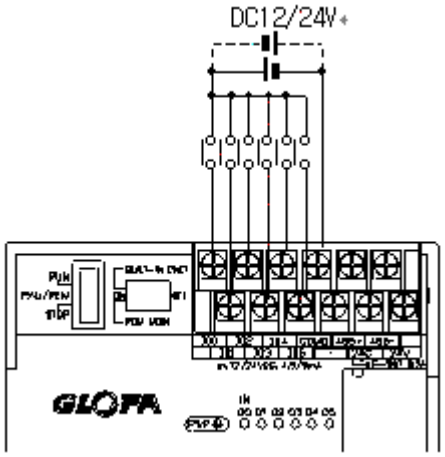
2) Circuit diagram



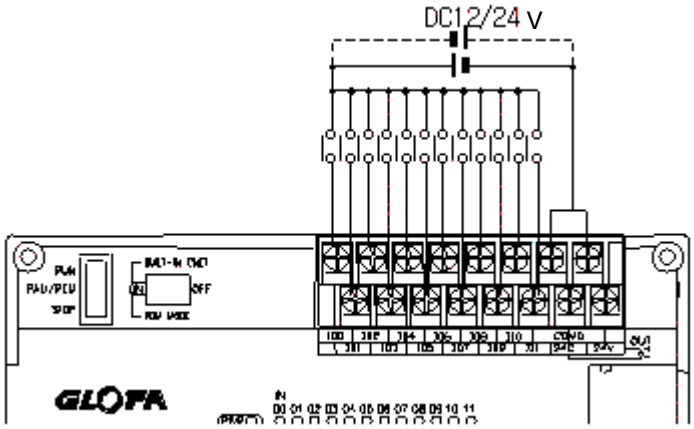
3) Input wiring

Base unit's wiring method is as follows. DC input specifications offered by GM7 is to be used for both electric current sink and electric current source.

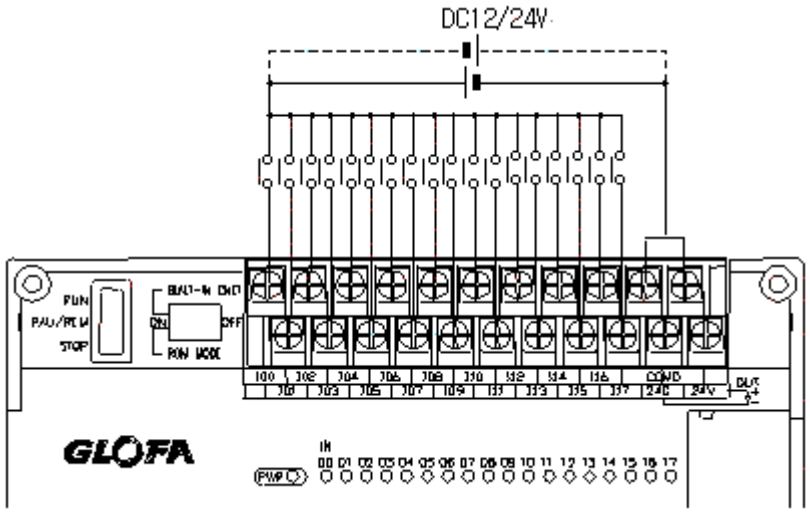
(1) 10-points base unit



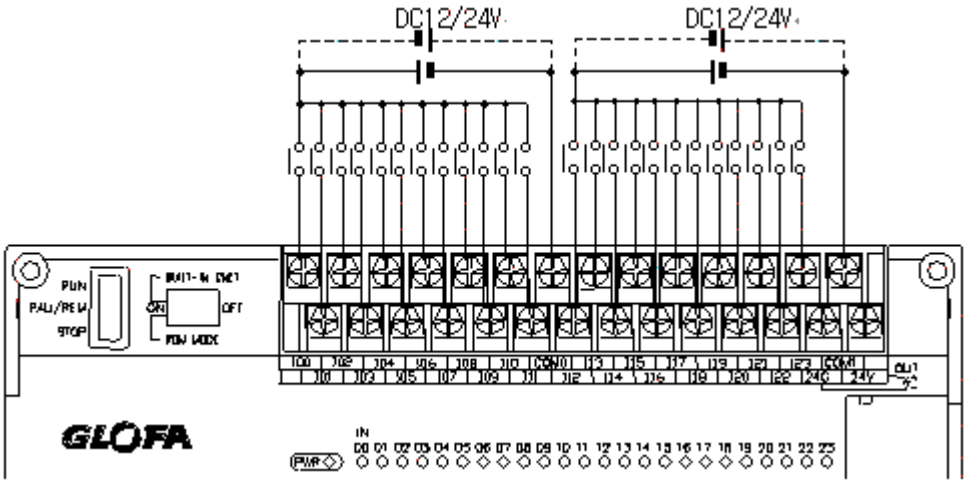
(2) 20-points base unit



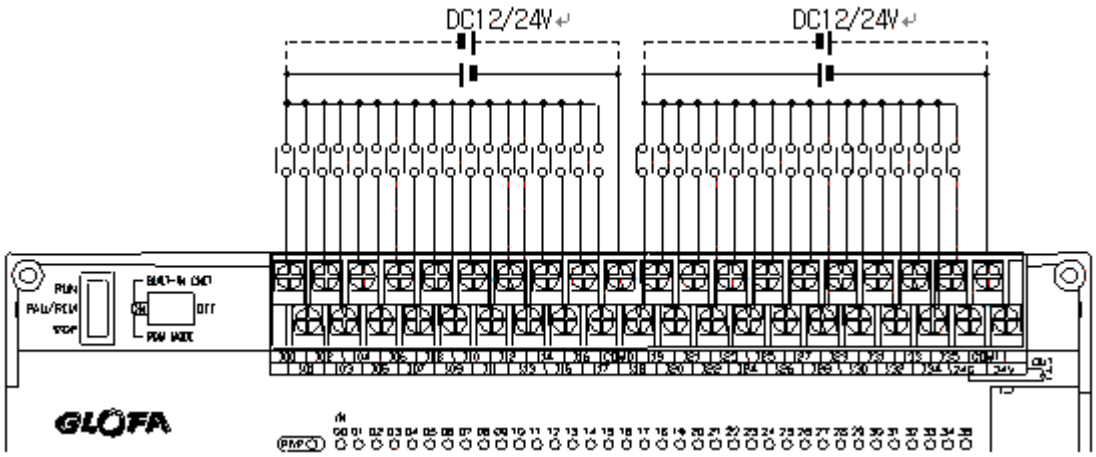
(3) 30-point base unit



(4) 40-point base unit

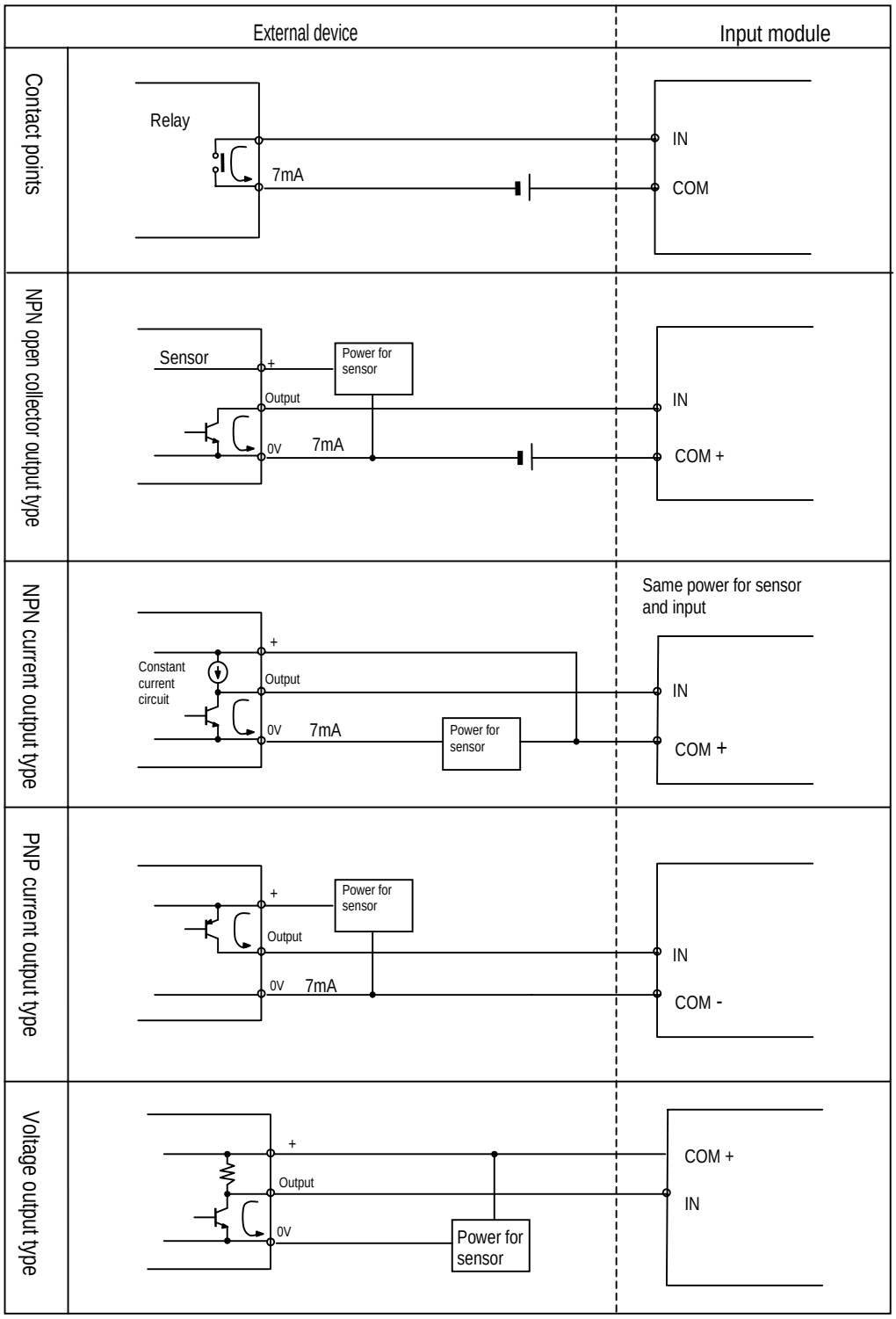


(5) 60-point base unit



4) Example of external devices.

To connect with external device of DC output type into DC input module, wire depending on the type of the external device as shown.



6.2.2 Expansion Module

1) Specifications

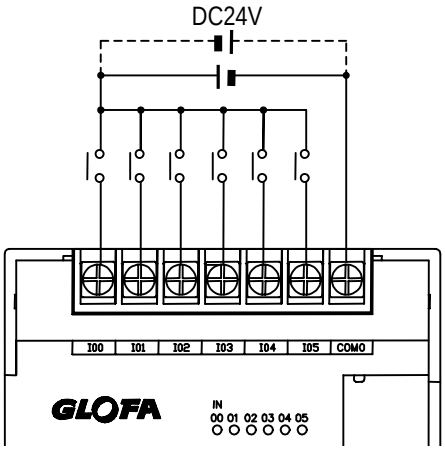
Model / Specification		Expansion Module
		G7E-DR10A
Number of input points		6 points
Insulation method		Photo coupler
Rated input voltage		DC 12 / 24V
Rated input current		4.5 / 9 mA
Operating voltage range		DC10.2 ~ 28.8V (ripple: less than 5%)
Max. Simultaneous input points		100% simultaneously On
On voltage / On current		DC9.5V or higher/ 4.3 mA or higher
Off voltage / Off current		DC5V or lower / 1.8 mA or lower
Input impedance		Approx. 2.7 kΩ
Response time	Off → On	15ms or less * 1
	On → Off	15ms or less * 1
Common terminal		6 points / com
Operating indicator		LED turns on at ON state of input

* 1: It's possible to select from 1ms to 15ms by 1ms at GMWIN.

2) Circuit diagram

It's the same with the one for the base unit.

3) Input wiring



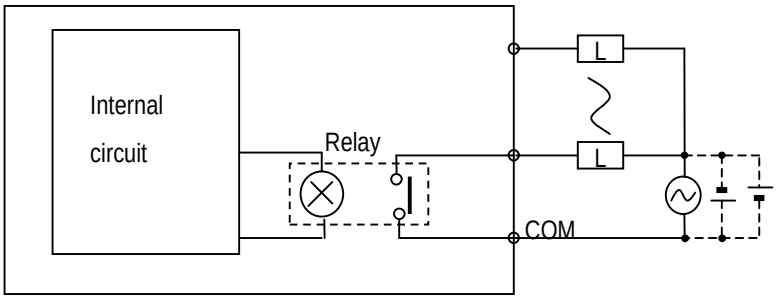
6.3 Digital Output Specification

6.3.1 Base unit (Relay output)

1) Specification

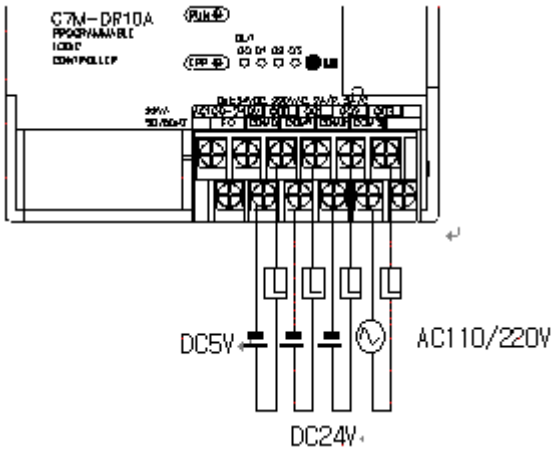
Model / Specifications		Base Unit				
		G7M-DR10A G7M-DR10A/DC	G7M-DR20A G7M-DR20A/DC	G7M-DR30A G7M-DR30A/DC	G7M-DR40A G7M-DR40A/DC	G7M-DR60A G7M-DR60A/DC
Output point		4 points	8 points	12 points	16 points	24 points
Insulation method		Relay insulation				
Rated load voltage/current		DC24V / 2A (r/load), AC220V / 2A (COS Ψ = 1) / 1 point 5A / 1COM				
Min. load Voltage/current		DC5V / 1mA				
Max. load voltage/current		AC250V, DC110V				
Current leakage when off		0.1mA (AC220V, 60Hz)				
Max. On/off frequency		1,200/hr				
Surge Absorber		None				
Life	Mechanical	More than 20,000,000				
	Electrical	Rated on/off voltage/current load 100,000 or more				
		AC200V / 1.5A, AC240V / 1A (COSΨ = 0.7) 100,000 or more				
		AC200V / 1A, AC240V / 0.5A (COSΨ = 0.35) 100,000 or more				
		DC24V / 1A, DC100V / 0.1A (L / R = 7ms) 100,000 or more				
Response time	Off → On	10 ms or less				
	On → Off	12 ms or less				
Common method		1 point/ 1COM, 2 points/ 1COM, 4 points/1COM				
Operation indication		LED is on at on status of output				

2) Circuit

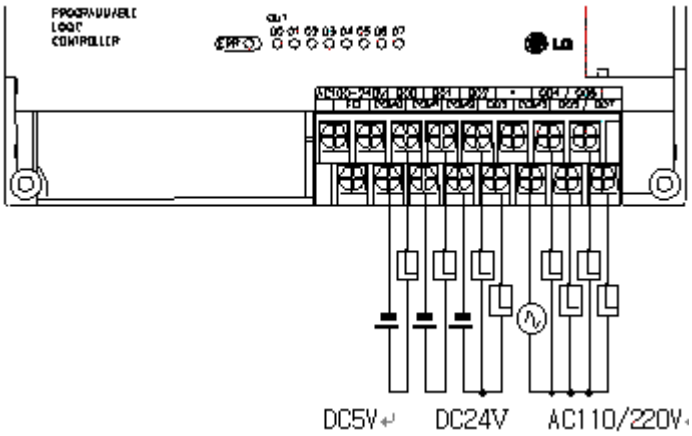


3) Output wiring

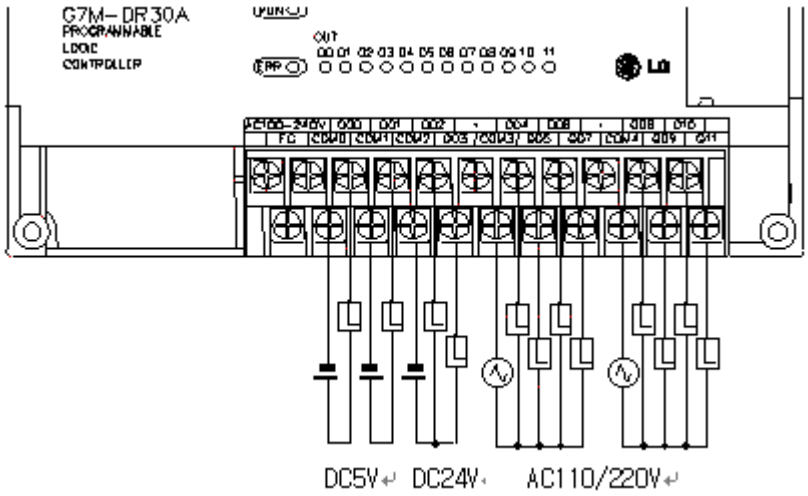
(1) 10-points base unit



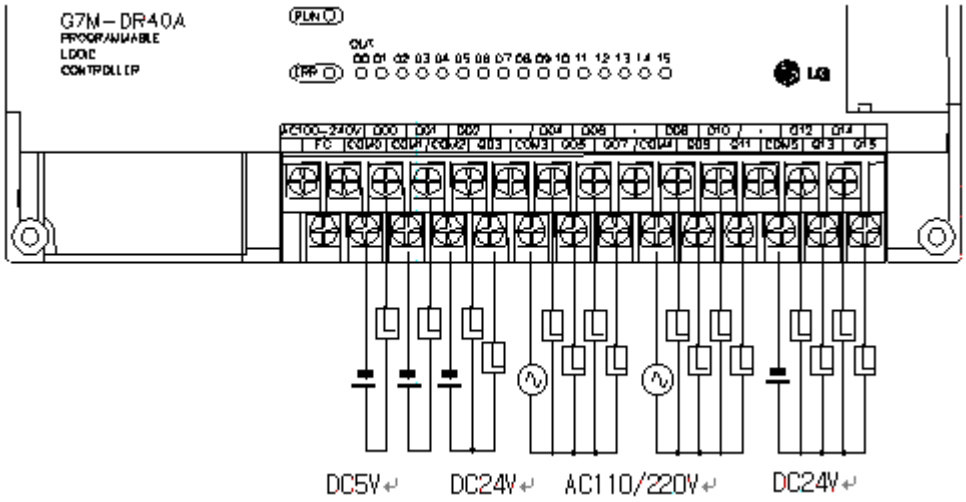
(2) 20-points base unit



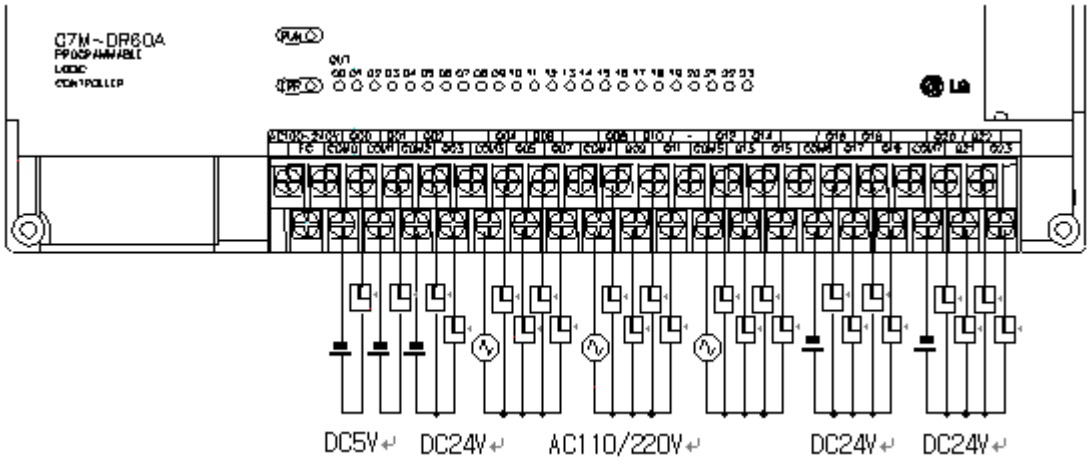
(2) 30-point base unit



(3) 40-point base unit



(4) 60-point base unit

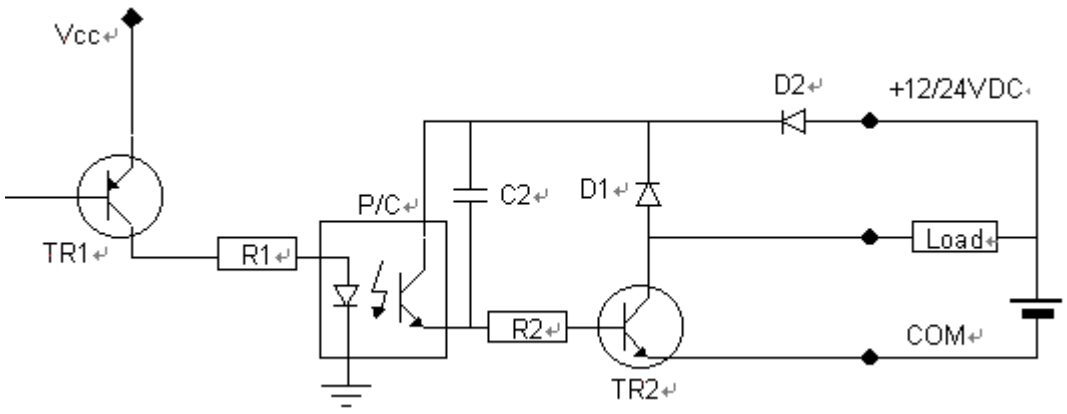


6.3.2 Base unit (Transistor output)

1) Specification

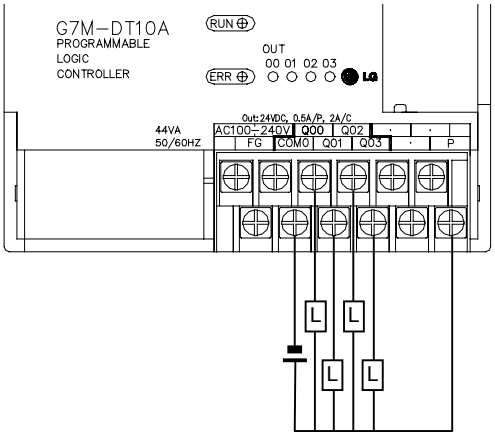
Model Specifications		Base Unit				
		G7M-DT10A	G7M-DT20A	G7M-DT30A	G7M-DT40A	G7M-DT60A
Output point		4 points	8 points	12 points	16 points	24 points
Insulation method		Photo coupler insulation				
Rated load voltage		DC12 / 24V, 0.5A / 1 point, 2A / 1COM				
Operating voltage		DC10.8 ~ 24.4V				
Max. load current		0.5A / 1 point, 3A / 1COM				
Current leakage when off		0.1mA or less at maximum load				
Max. voltage drop when on		DC1.5V or less				
Max. inrush current		4A / 10ms or less				
Surge Absorber		Clamp diode				
Response time	Off → On	2 ms or less				
	On → Off	2 ms or less				
External power supply	Voltage	DC24V ± 10% (Ripple voltage : 4 Vp-p or less)				
	Current	20mA or less	30mA or less	40mA or less	50mA or less	80mA or less
Common method		4 point / 1COM	8 points / 1COM	8 points / 1COM 4 points / 1COM	8 points / 1COM (x2)	8 points / 1COM (x3)
Operation indication		LED is on at on status of output				

2) Circuit

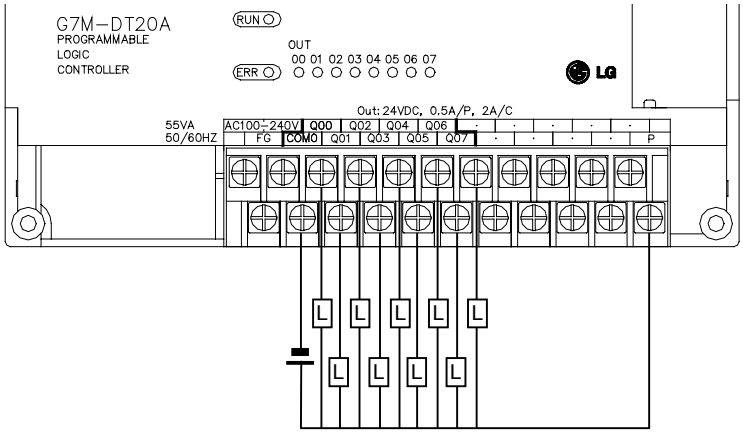


3) Output wiring

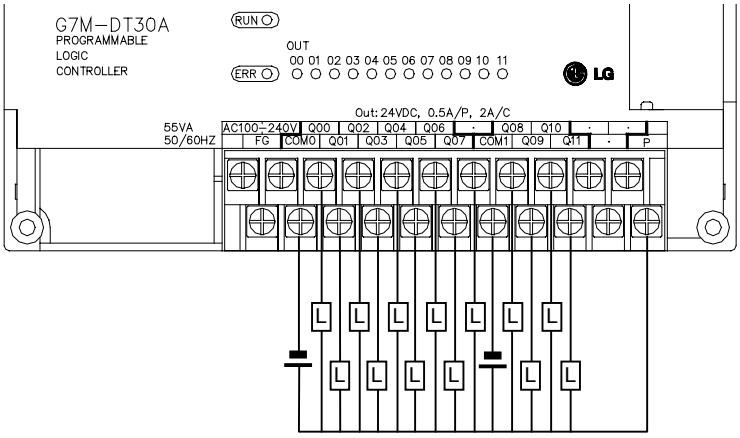
(1) 10-points base unit



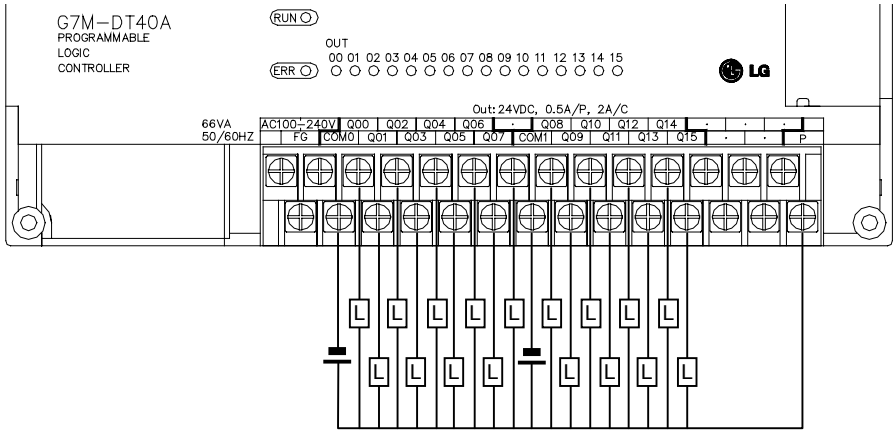
(2) 20-points base unit



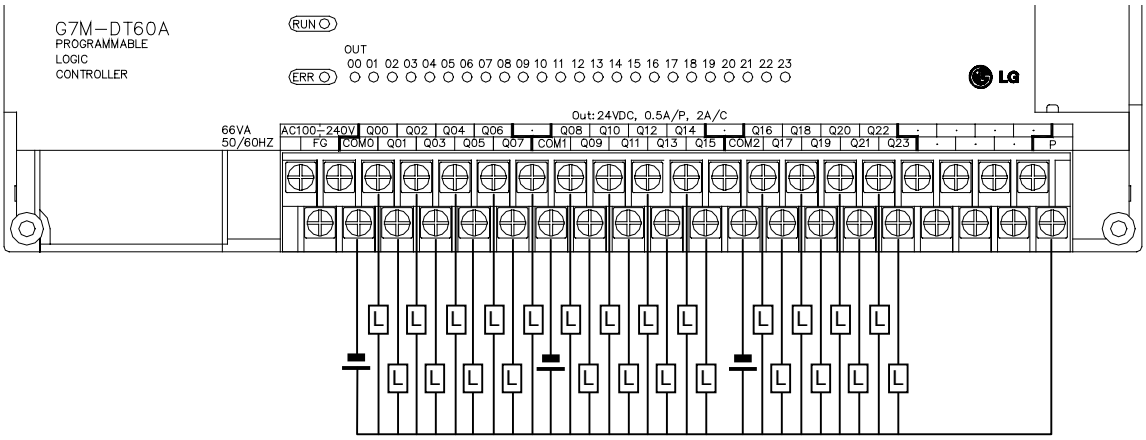
(2) 30-point base unit



(3) 40-point base unit



(4) 60-point base unit



6.3.2 Expansion Module

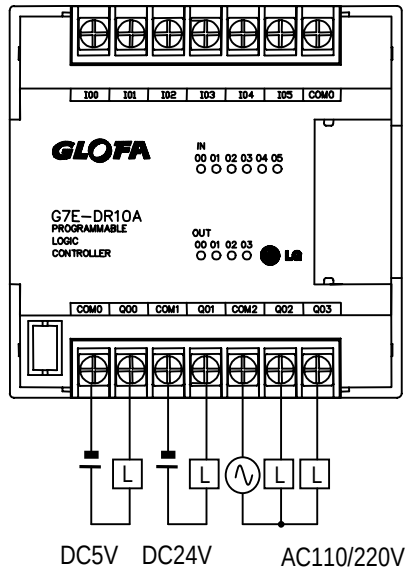
1) Specifications

Specifications \ Model		Expansion Module
		G7E-DR10A
Output point		4 points
Insulation method		Relay insulation
Rated load Voltage/current		DC24V / 2A (r/load), AC220V / 2A (COS Ψ = 1) / 1 point 5A / 1COM
Min. load Voltage/current		DC5V / 1mA
Max. load voltage/current		AC250V, DC110V
Current leakage when off		0.1mA (AC220V, 60Hz)
Max. On/off frequency		1,200/hr
Surge Absorber		None
Life	Mechanical	More than 20,000,000
	Electrical	Rated on/off voltage/current load 100,000 or more
		AC200V / 1.5A, AC240V / 1A (COSΨ = 0.7) 100,000 or more
		AC200V / 1A, AC240V / 0.5A (COSΨ = 0.35) 100,000 or more
		DC24V / 1A, DC100V / 0.1A (L / R = 7ms) 100,000 or more
Response time	Off → On	10 ms or less
	On → Off	12 ms or less
Common method		1 point/ 1COM, 2 points/ 1COM
Operation indication		LED is on at on status of output

2) Circuit

It's the same with the output circuit of the base unit.

3) Output wiring



REMARK

1) Refer to 7. 2 'Special Functions' for the special function units

Chapter 7 Usage of Various Functions

7.1 Built-in Functions

7.1.1 High-speed counter function

This chapter describes the specification, handling, and programming of built-in high speed counter of GM7. The built-in high speed counter of GM7 (Hereafter called HSC) has the following features;

3 counter functions as followings
- 1-phase up / down counter : Up / down is selected by user program
- 1-phase up / down counter : Up / down is selected by external B phase input
- 2-phase up / down counter : Up / down is automatically selected by the phase difference between A-phase and B.

Multiplication (1, 2, or 4) with 2-phase counter
- 2-phase pulse input multiplied by one : Counts the pulse at the leading edge of A-phase.
- 2-phase pulse input multiplied by two : Counts the pulse at the leading / falling edge of A-phase.
- 2-phase pulse input multiplied by four : Counts the pulse at the leading / falling edge of A-phase and B

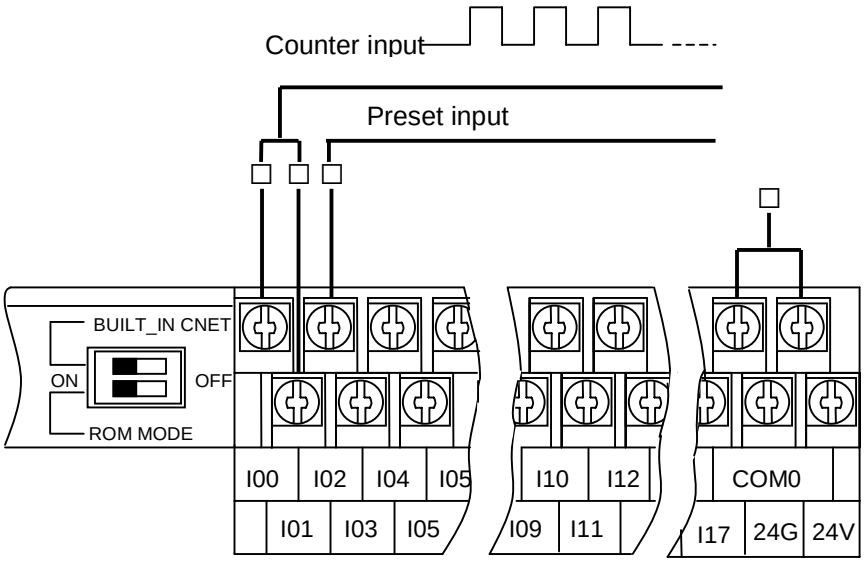
1) Performance Specifications

Items		Specifications
Input signal	Types	A-phase, B-phase, Preset
	Rated level	24VDC (15mA)
	Signal type	Voltage input
Counting range		0 ~ 16,777,215 (Binary 24 bits)
Max. counting speed		1-phase 16kHz/ 2-phase 8kHz
Up / Down selection	1-phase	Sequence program or B-phase input
	2-phase	Auto-select by phase difference of A-phase and B
Multiplication		1, 2, or 4
Preset input		Sequence program or external preset input

2) Input specification

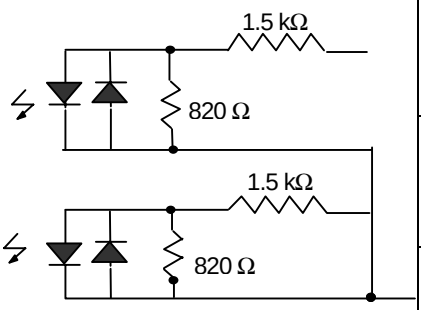
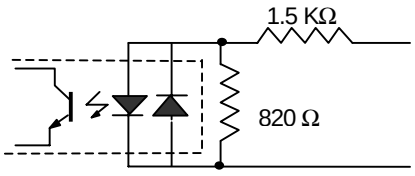
Items		Specifications
A / B phase	Rated input	24VDC (15mA)
	On voltage	14VDC or higher
	Off voltage	2.5VDC or lower
Preset input	Rated input	24VDC (15mA)
	On voltage	19VDC or higher
	Off voltage	6V or lower
	On delay time	Less than 1.5ms
	Off delay time	Less than 2ms

3) Names of wiring terminals



No.	Terminal No.	Names	Usage
□	I00	ϕ A 24V	A Phase input terminal
□	I01	ϕ B 24V	B Phase input terminal
□	I02	Preset 24V	Preset input terminal
□	COM0	Common input	Common terminal

4) External interface circuit

I/O	Internal circuit	Terminal No.	Signal name	Operation	Input warranted voltage
Input		I00	A-phase pulse Input (DC24V)	On	14 ~ 26.4 V
				Off	2.5V□□
		I01	B-phase pulse Input (DC24V)	On	14 ~ 26.4 V
				Off	2.5V or lower
		COM0	COM (input common)	——	
Input		I02	Preset input (DC24V)	On	19 ~ 26.4 V
				Off	6V or lower
		COM0	COM (input Common)	——	

5) Wiring instructions

A high speed pulse input is sensitive to the external noise and should be handled with special care. When wiring the built-in high speed counter of GM7, take the following precautions against wiring noise.

(1) Be sure to use shielded twisted pair cables. Also provide Class 3 grounding.

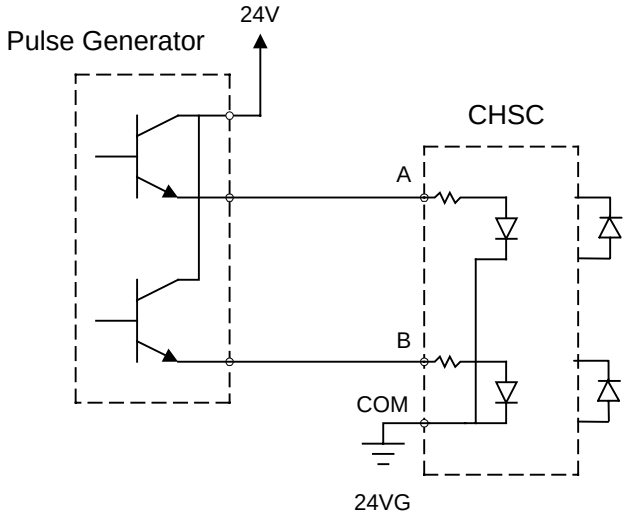
(2) Do not run a twisted pair cable in parallel with power cables or other I/O lines which may generate noise.

(3) Before applying a power source for pulse generator, be sure to use a noise-protected power supply.

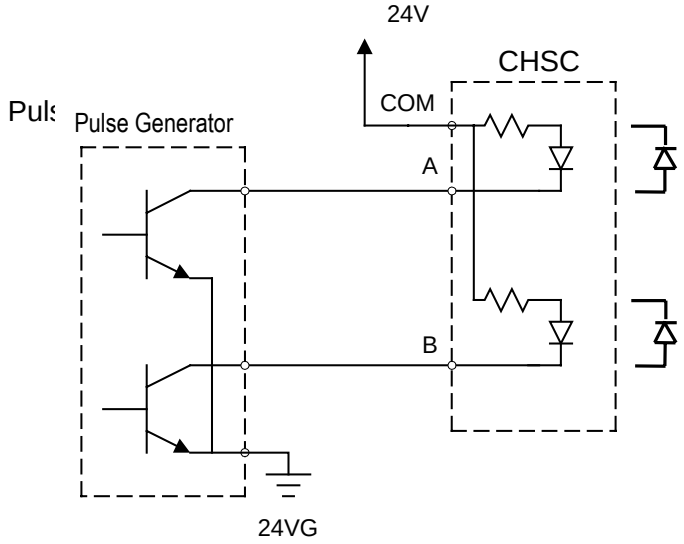
(4) For 1-phase input, connect the count input signal only to the phase A input; for 2-phase input, connect to phases A and B.

6) Wiring example

(1) Voltage output pulse generator

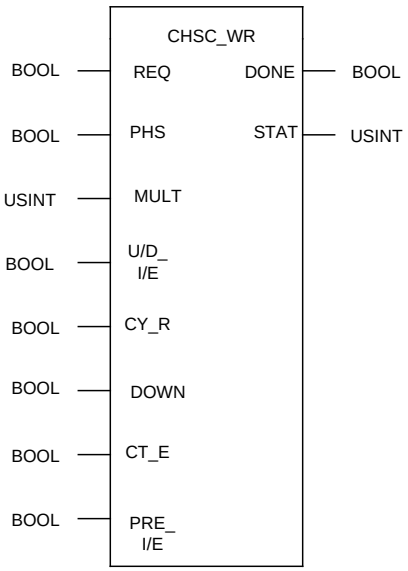


(2) open collector output pulse genernator



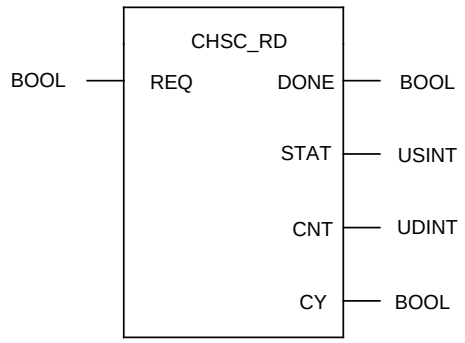
7) Function block

(1) Writing the operating information(CHSC_WR)

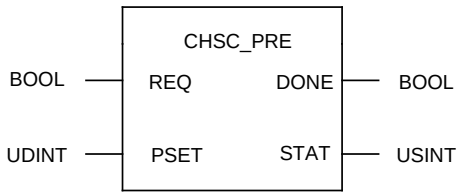
FUNCTION BLOCK	Description
	Input REQ : Request signal of F/B execution PHS : Operation modes selection 0 (1-phase counter), 1(2-phase counter) MULT : Assign the multiplication factor (MULT = 1, 2, or 4) U/D_I/E : Assign the count direction(up/down) selector 0 : Set by sequence program 1 : Set by B-phase input signal (1:up-count, 0:down-count) CY_R : Carry reset signal (1: reset). DOWN : Select the count direction (0:up/1:down) when the counter is set as 1-phase counter and up/down is selected by sequence program. (PHS=0 & U/D_I/E=0) CT_E : Counter enable signal (0 : Counter disable, 1 : Enable) PRE_I/E : Assign PRESET input 0 : PRESET by sequence program 1 : PRESET by external input at the PRESET terminal Output DONE : Turns on after the F/B is executed with no error. STAT : Indicate the operation status of F/B

- The MULT input will be dummy input when the HSC is set as 1-phase counter (PHS = 0). When the HSC is set as 2-phase counter, the U/D_I/E and DOWN input will be dummy input. (PHS = 1)
- The current value of HSC will be cleared as 0 when the CT_E (counter enable) is 0.

(2) Reading the current value and operation status of HSC(CHSC_RD)

FUNCTION BLOCK	Description
	**Input** REQ : Request signal for F/B execution **Output** DONE : Turns on after the F/B is executed with no error. STAT : Indicates the operation status of F/B CNT : The current value of HSC (0 ~ 16,777,215) CY : Carry flag (0 : OFF, 1 : ON)

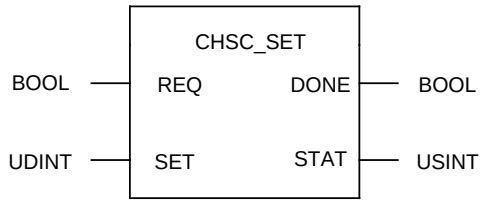
(3) Set the preset value of HSC(CHSC_PRE)

FUNCTION BLOCK	Description
	**Input** REQ : Request signal for F/B execution PSET : Set the preset value (0 ~ 16,777,215) **Output** DONE : Turns on after the F/B is executed with no error. STAT : Indicates the operation status of F/B

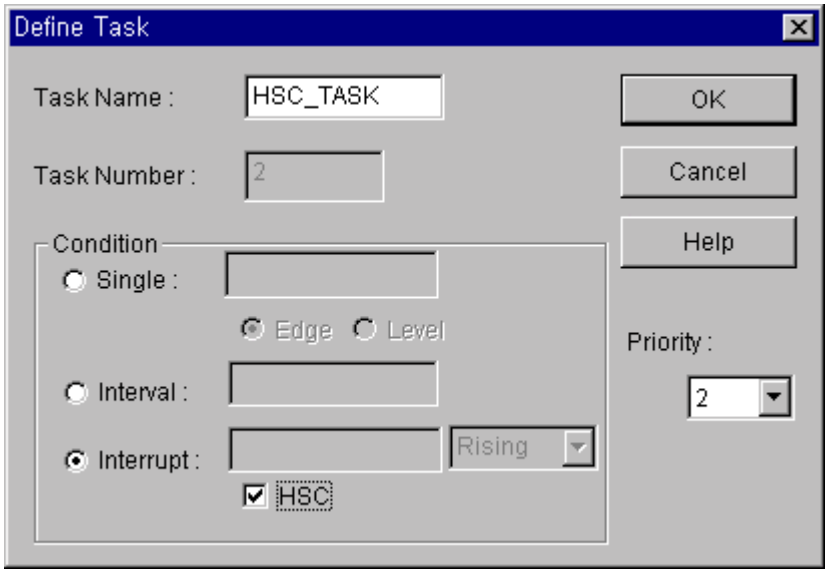
- When the PRE_I/E is set as 0 (Preset input by sequence program), the current value of HSC is changed as the signed preset value with the rising edge of REQ input.
- When the PRE_I/E is set as 1 (Preset input by external preset input), the current value of HSC is changed as the assigned preset value with the rising edge of external preset input. At this time, the REQ input of CHSC_PRE is ignored.
- The CY output is set off while the CHSC_PRE F/B is executing.
- The CHSC_PRE F/B is disabled while the CT_E input of CHSC_WR F/B is 0 (Counter disabled).

(4) Setting up comparison value

Assign a setting value to be compared with the current of HSC.

FUNCTION BLOCK	Description
	**Input** REQ : Request signal for F/B execution SET : Set a setting value (0 ~ 16,777,215) **Output** DONE : Turns on after the F/B is executed with no error. STAT : Indicates the operation status of F/B

Run a task program when the current value of HSC reaches to the setting value.
To run a task program, define a high speed counter task program as following figure, and write a task program.



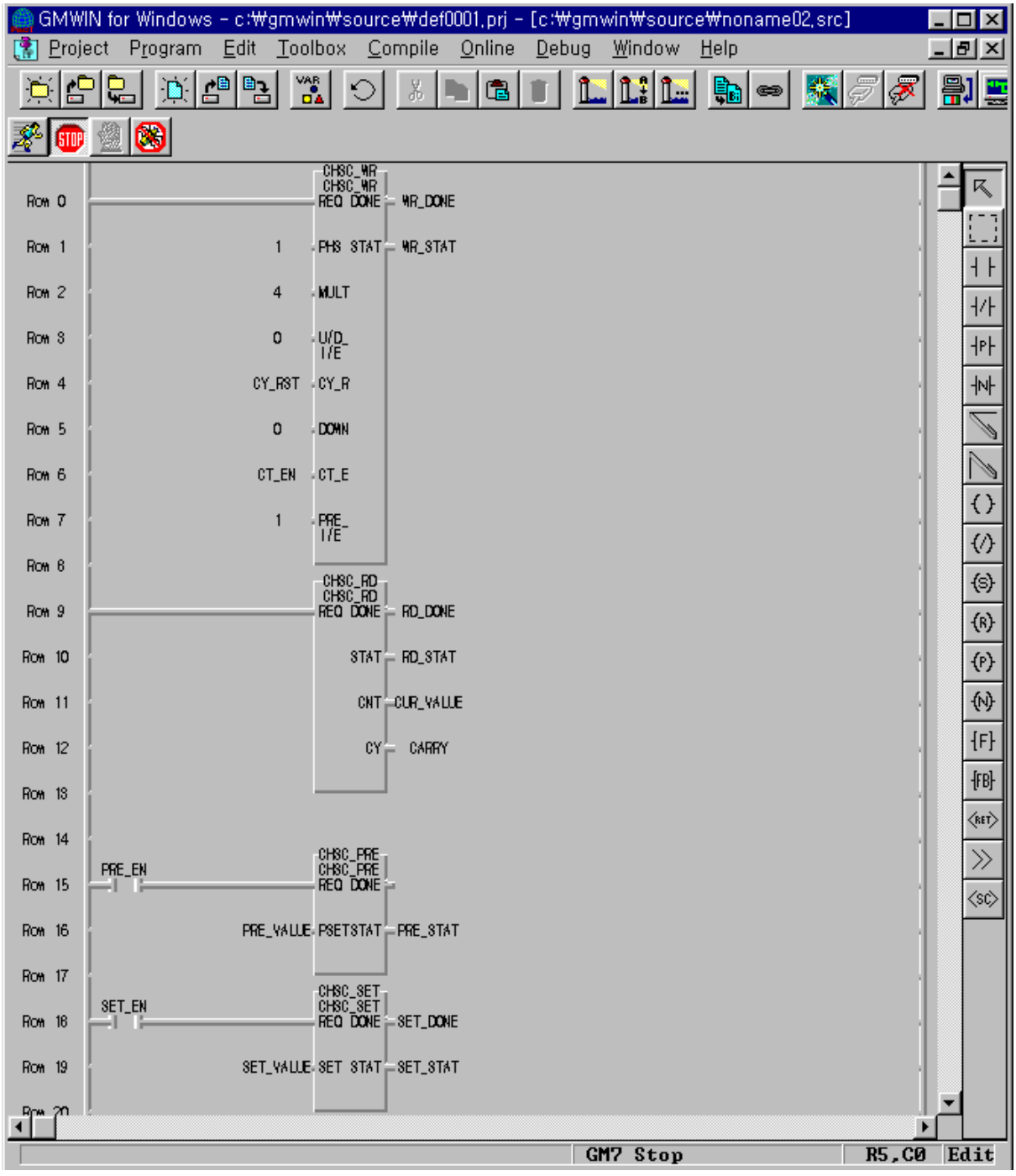
(5) Error code of F/B

The following table shows error codes appear at the STAT output.

Error code	Description
00	No error
01	-
02	Input data error at MULT input of CHSC_WR (When the number is not one of 1, 2, 4 on the 2-Phase Mode)
03	PSET (CHSC_PRE) or SET (CHSC_SET) is out of specified range (0 ~ 16,777,215).
04	Execute Preset command while the HSC is disabled status

8) Example of the program

This is an 4 multiplication example when the preset is designated by the external preset input.

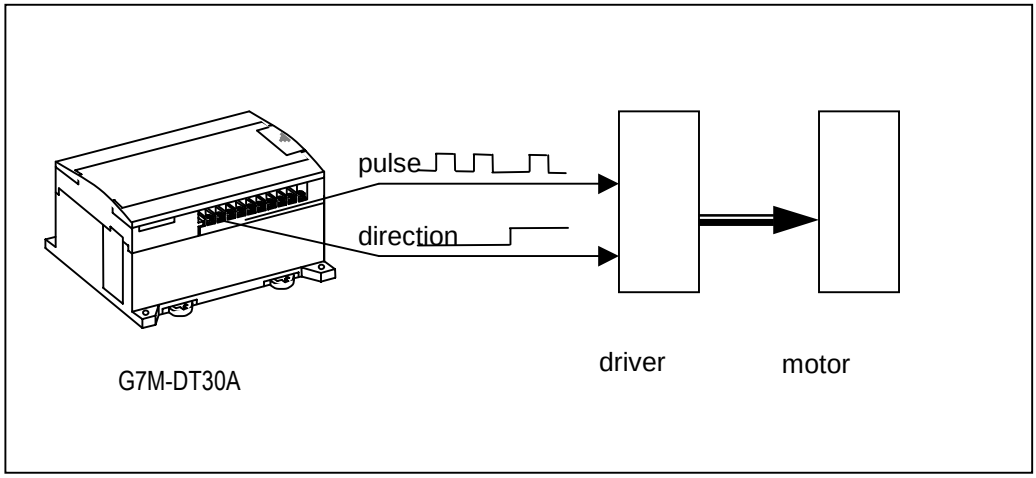


7.1.2. Pulse Output Function

In the transistor output type of GM7, the pulse output function - maximum 2Kpps - is internalized. By using this function with stepping motor or servo motor driver, GM7 is applicable to a simple positioning system.

1) Usage of the Pulse Output

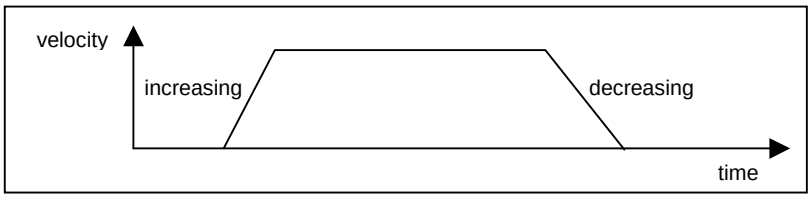
Transistor output type of GM7 outputs the signals of pulse and direction in an output contact point through the function block of the pulse output (PULS_OUT). The outputted pulse is connected to motor driver it is controlled position in the following figure.



Choose a mode from the pulse out function block (PULS_OUT) and operate following 3 modes

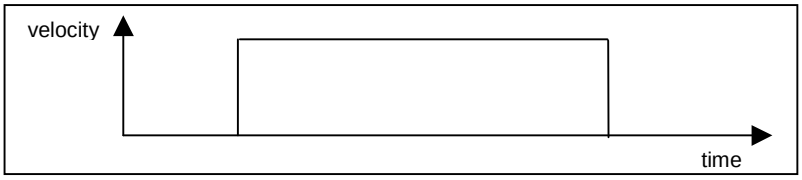
(1) Trapezoidal operation

The pulse output function operates in order of acceleration – uniform velocity – deceleration.



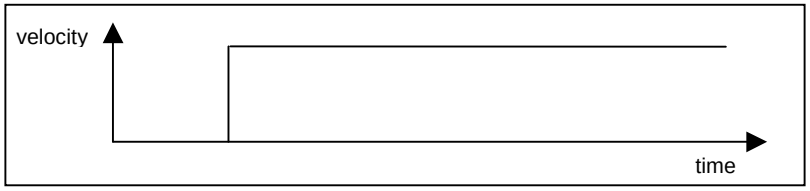
(2) Uniform velocity operation

Operates with the uniform velocity without increasing/decreasing operation



(3) Infinite operation

Operate infinitely without an increasing/decreasing operation until meet the emergency stop command.



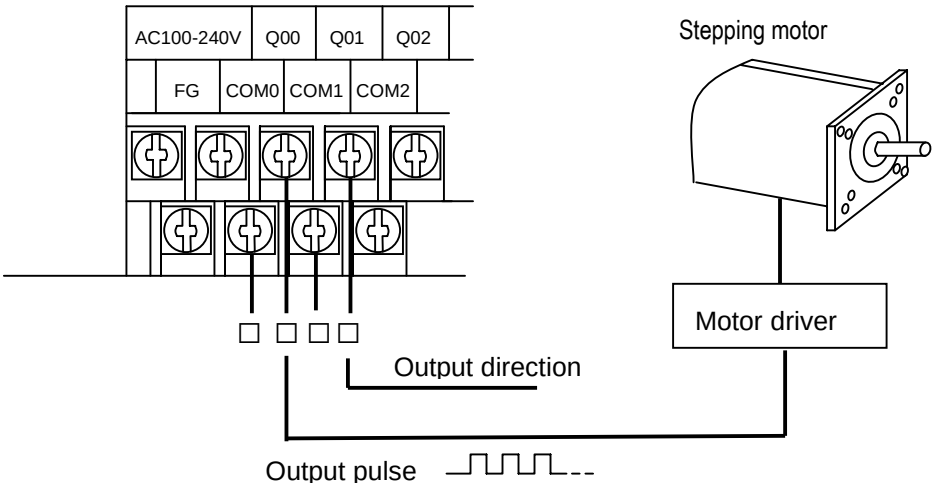
2) Functional Specification

Item	Specification
No. of output	1 point
Output type	Pulse
Output velocity	Max 2Kpps, Min 50pps
Output pulse	0 ~ 2147483647
Execution type of the increasing/decreasing velocity	Designation of acceleration
Type of the direction designation	Right/opposite direction pulse output
Load power supply	DC 12V/24V
Usable range of the load power supply	DC10.2 ~ 26.4V
Maximum load current	150mA
Initiative electric current	Less than 0.4A, 10ms
Maximum power dropdown under On	Less than DC 0.5V
Electric current leakage under Off	Less than 0.1mA
On delayed time	Less than 1ms
Off delayed time	Less than 1ms

Remark

1) Several points can be used for the pulse output point if they are not output at the same time. Thus it is possible that right direction pulse is output as %QX0.0.0, opposite direction pulse is output as %QX0.0.1.

3) Names of parts

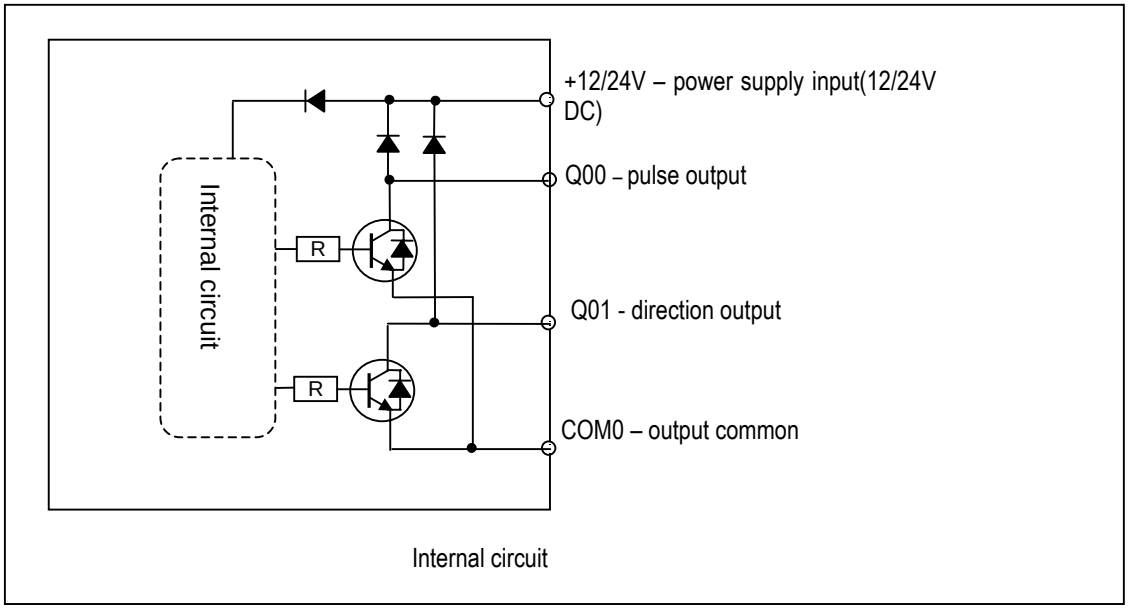


No.	Terminal No.	Names	Usage
□	Q00	Pulse output	Pulse output terminal of right direction
□	COM0	Common	Pulse output common terminal
□	Q01	Direction output	Direction output terminal
□	COM0	Common	Direction output common terminal

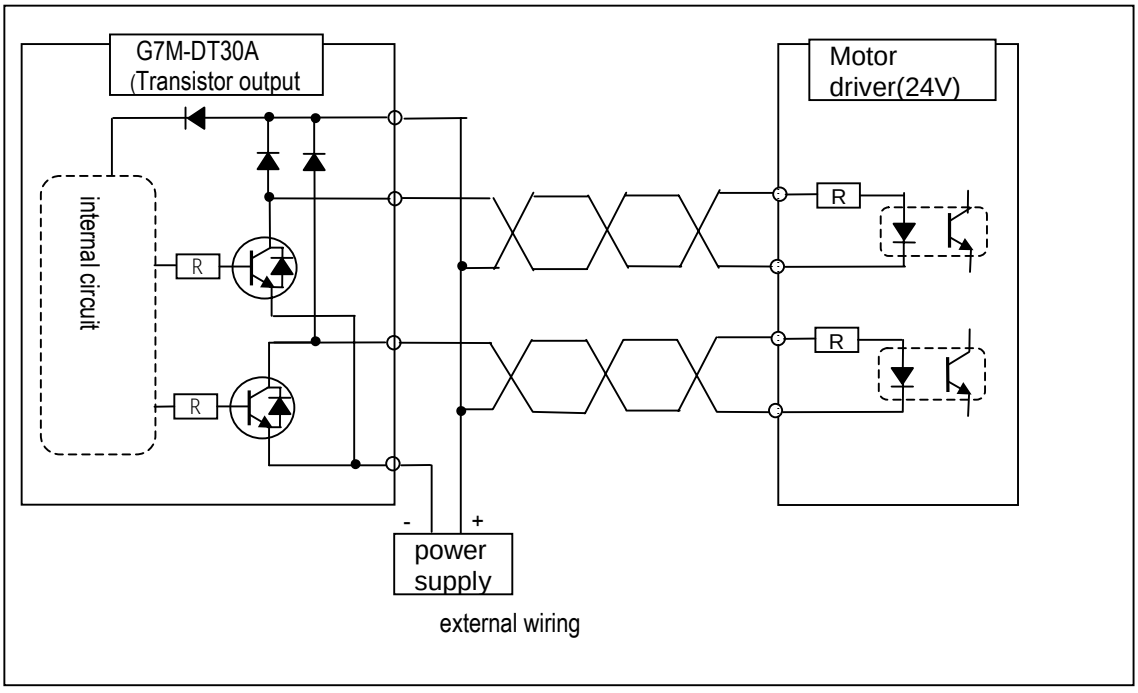
Remark

If the motor drive is not input direction, but is input right/opposite direction pulse (the opposite direction pulse can be output through using 2 function blocks (PULS_OUT) to Q01 contact point

4) Internal circuit and external wiring



Internal circuit



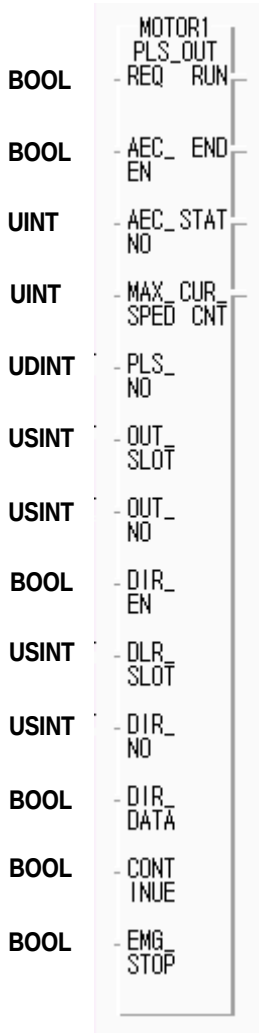
external wiring

Remark

Be careful about the counter plan of the noise during the wiring in the pulse output.
1) Use twisted pair shields wire for wiring and execute 3rd contact point.
2) Be sure to separate from the power supply line and I/O lines on which noise usually occurs.
3) Length of wire should be as short as possible.
4) Be sure to use the stable power supply for the pulse output and separate it from I/O power supply.

5) Function Block

(1) Pulse Output (PLS_OUT) Function Block

Function Block	Explanation
	**Input** REQ: Request to execute the function block (operating by rising edge) AEC_EN: designation of increasing/decreasing velocity operation 0: uniform velocity operation 1: increasing/decreasing velocity operation AEC_NO: Increasing/decreasing operating is direct proportion to the ratio of the inclination I/O velocity and setting up value *6) refers to operating exp. MAX_SPED: output cycle of the max. Pulse Range of setting up: 50~2000(pps) Can be set up by the multiple of 50 PLS_NO: No. of the pulse output Range of setting up: 0~4294967295 OUT_SLOT: designation slot number contact point for the pulse output OUT_NO: designation number of contact point for the pulse output DIR_EN: designation whether the usage of direction output or not. 0: No direction output 1: using direction output *6) refers to operating exp. DIR_SLOT: Designation slot number of contact point for direction signal output DIR_NO: Designation number of contact point for direction signal output DIR_DATA: Direction output data 0: right direction 1: opposite direction CONTINUE: Designation infinite operation 0: uniform velocity or I/D velocity operation 1: Infinite operation before stop order EMG_STOP: Stop order Immediate stop without decreasing velocity while EMG_STOP goes 1 during the operation **Output** RUN: remark status of pulse output 0: during stop 1: during pulse output END: remark status of operation 0: during pulse output 1: during stop CUR_CNT: remark no. of current pulse output

(2) Function Block Error List

Error status	Contents	Treatment
00	Normal	-
01	Other PLC_OUT F/B's pulsating.	Change the other PLC_OUT F/B's program.
02	Velocity designation error (more than 2000, not a multiple of 50, designated 0)	Velocity designation adjustment
03	The no. of I/D velocity pulse is bigger than no. of all pulse is to output.	Acceleration adjustment
04	No output contact point where is designated to the pulse output	Output contact point designation
05	No output contact point where is designated to the direction output	Output contact point designation

6) Operating explanation

(1) Increasing/Decreasing velocity inclination

It can be adapted for initiative point and end point of pulse output operation.
- Increasing velocity inclination: inclination, which is from velocity 0 to designated max. Output cycle.
- Decreasing velocity inclination: inclination, which is from designated, limited value of velocity-to-velocity 0.

Remark

Increasing velocity inclination and decreasing velocity inclination of GM7 pulse output are set up as the same. Set up proper value by the sort of motor because if I/D velocity increases, the arrival time to the designated max. Cycle also increases.

Condition 1)

Set up as AEC_NO = 1, MAX_SPED = 1000, PLS_NO = 5000.

I/D inclination is 1, output cycle of max. Pulse is 1000pps and the no. Of the output pulse Is 5000.

① If I/D velocity inclination is 1, 1 pulse is output on the 1st step (velocity: 50pps).
Pulse velocity is 50pps, so time consuming is 20ms.

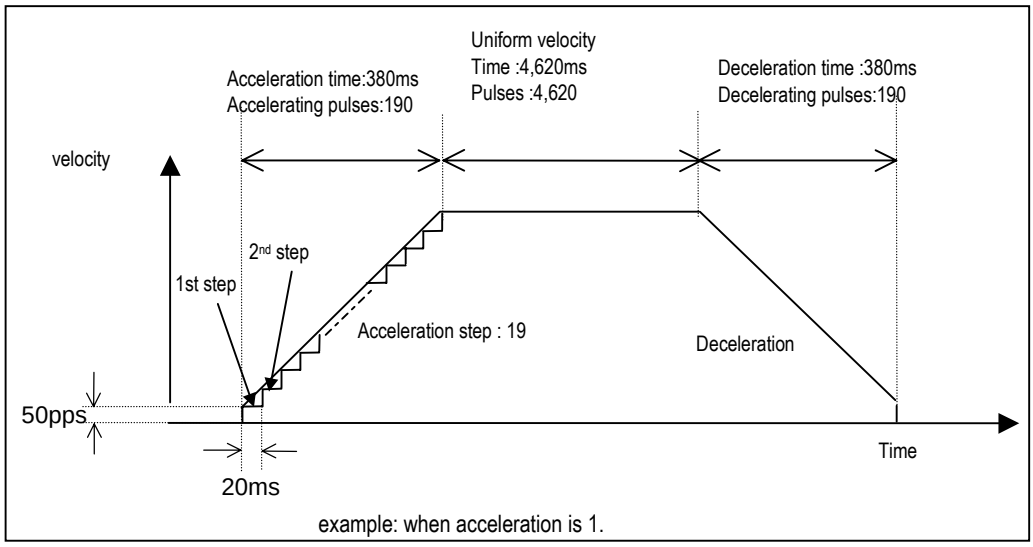
② 2 pulses are output on the 2nd step (velocity: 100pps) and time consumes 20ms

③ By calculation in the same way, the time to reach to 1000pps is
20ms * (20-1) = 380ms, and the no. of output pulses are 1+2+3...+18+19 = 190 units.

④ Decreasing velocity inclination is 1, thus 190 units of pulses are needed.

⑤ The no. of pulses in the uniform velocity region are 5000-190-190=4,620 units.

⑥ Whole spent time is 50,380ms



example: when acceleration is 1.

Condition 2

Set up as AEC_NO = 2, MAX_SPED = 1000, PLS_NO = 5000.

I/D inclination is 2, output cycle of max. pulse is 1000pps and the no. of the output pulse is 5000.

① If I/D velocity inclination is 2, 2 pulses are output on the 1st step(velocity: 50pps). Pulse velocity is 50pps. So time consuming is 40ms.

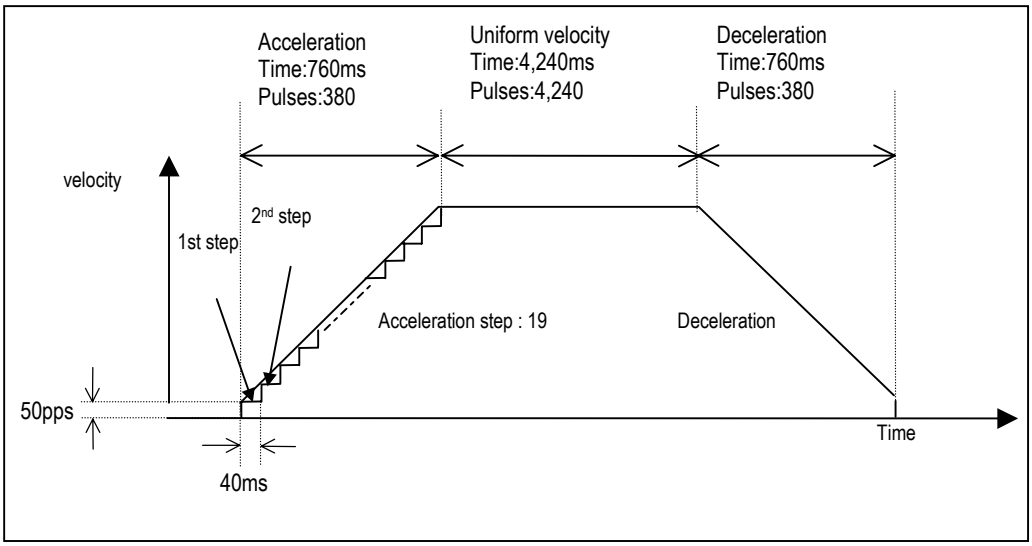
② 4 pulses are output on the 2nd step(velocity: 100pps) and time consumes 20ms

③ By calculation in the same way, the time to reach to 1000pps is 40ms * (20-1) = 760ms, and the no. of output pulses are 2+4+6...+36+38 = 380 units.

④ Decreasing velocity inclination is 2, thus 380 units of pulses are needed.

⑤ The no. of pulses in the uniform velocity region are 5000-380-380=4,240 units.

⑥ Whole spent time is 57,600ms



Example) Acceleration is 2.

Remark

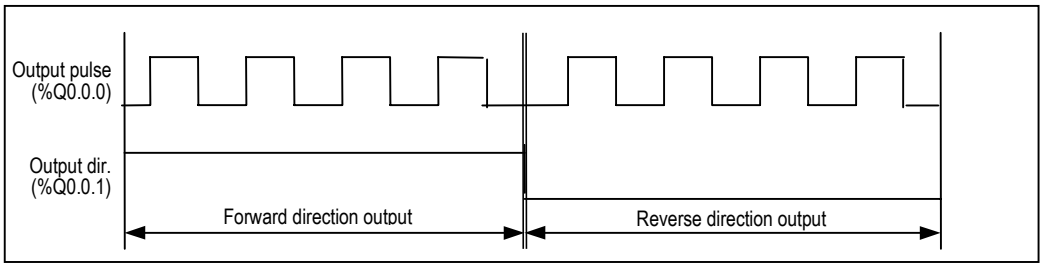
If the increasing velocity inclination goes bigger, the increasing time and pulse go bigger by direct proportion to inclination. Then be careful of an occurring of the function block error when the no. of I/D velocity pulse becomes bigger than the no. of whole pulse.

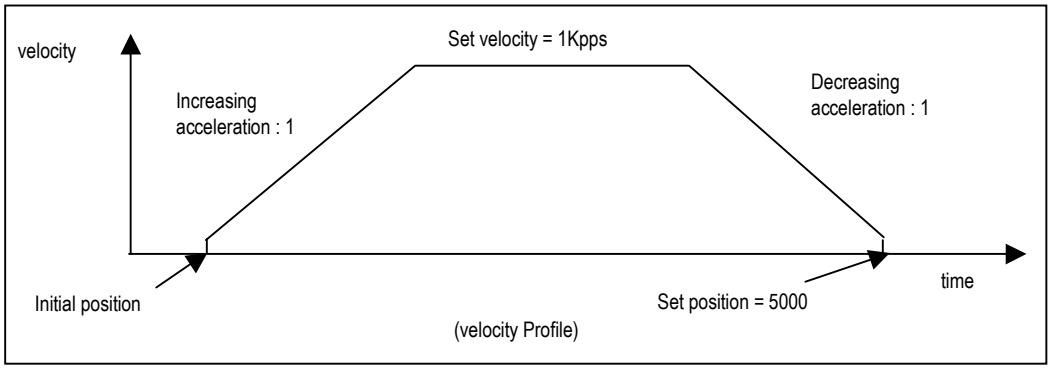
(2) Output Direction

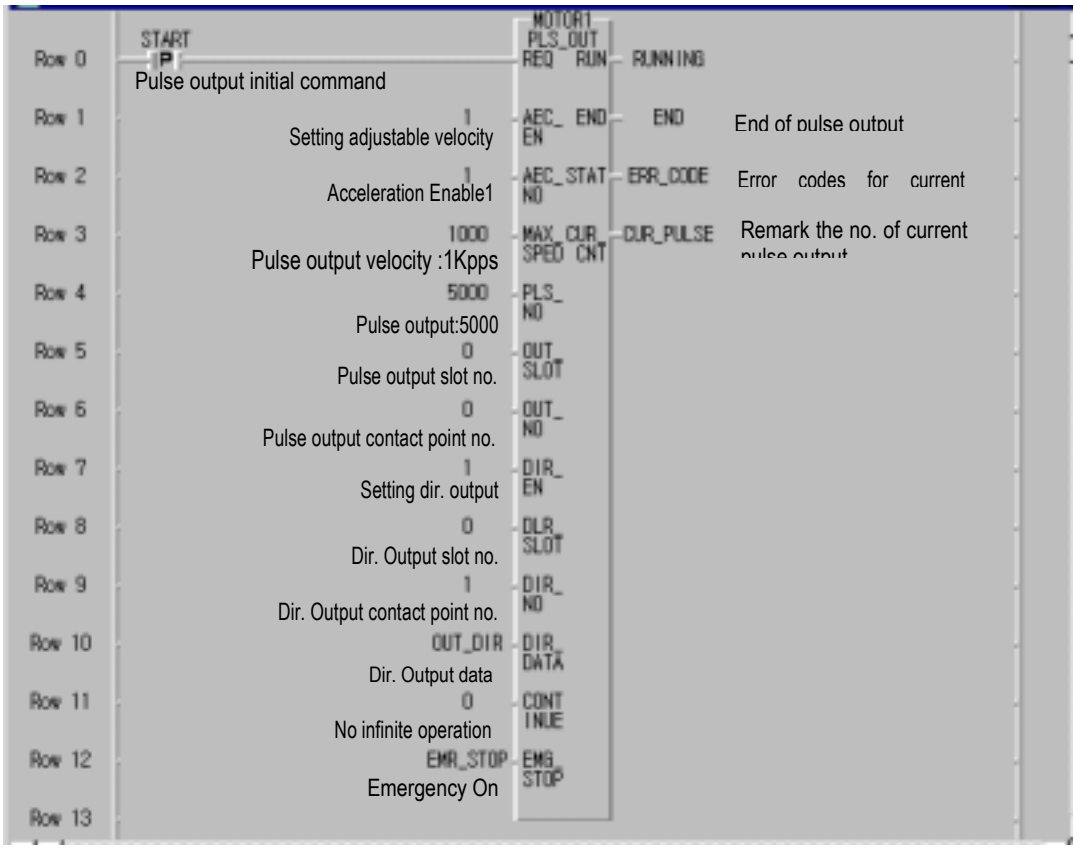
Input type of servo motor driver or stepping motor driver is subdivided into 2. Output direction of control can be selected in the pulse output function block.

a) Selecting method of output direction

(a) When driver gets input forward direction pulse and reverse direction pulse contact point, and the forward/reverse direction signals one levels.

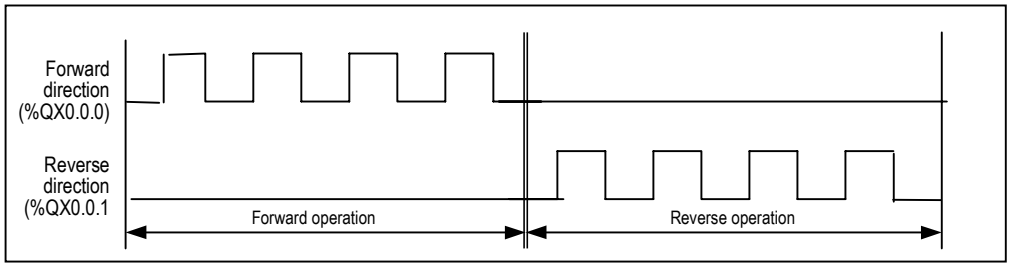


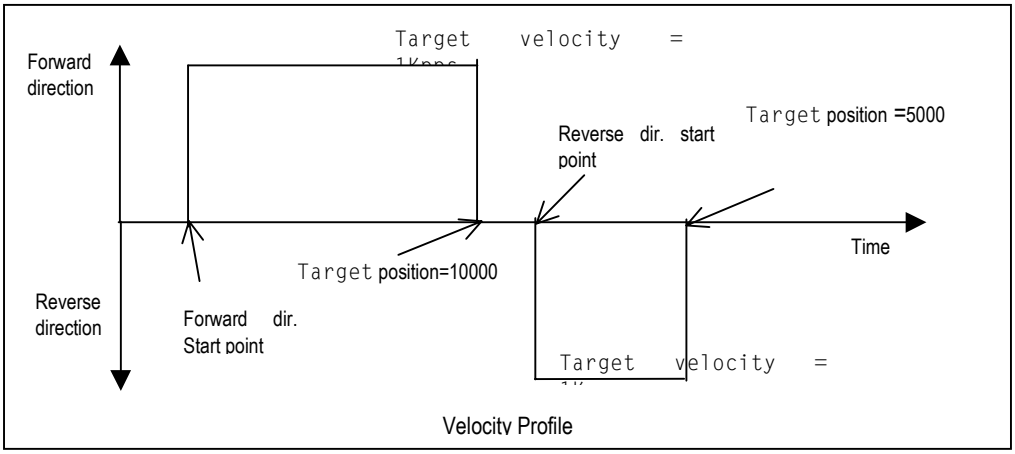




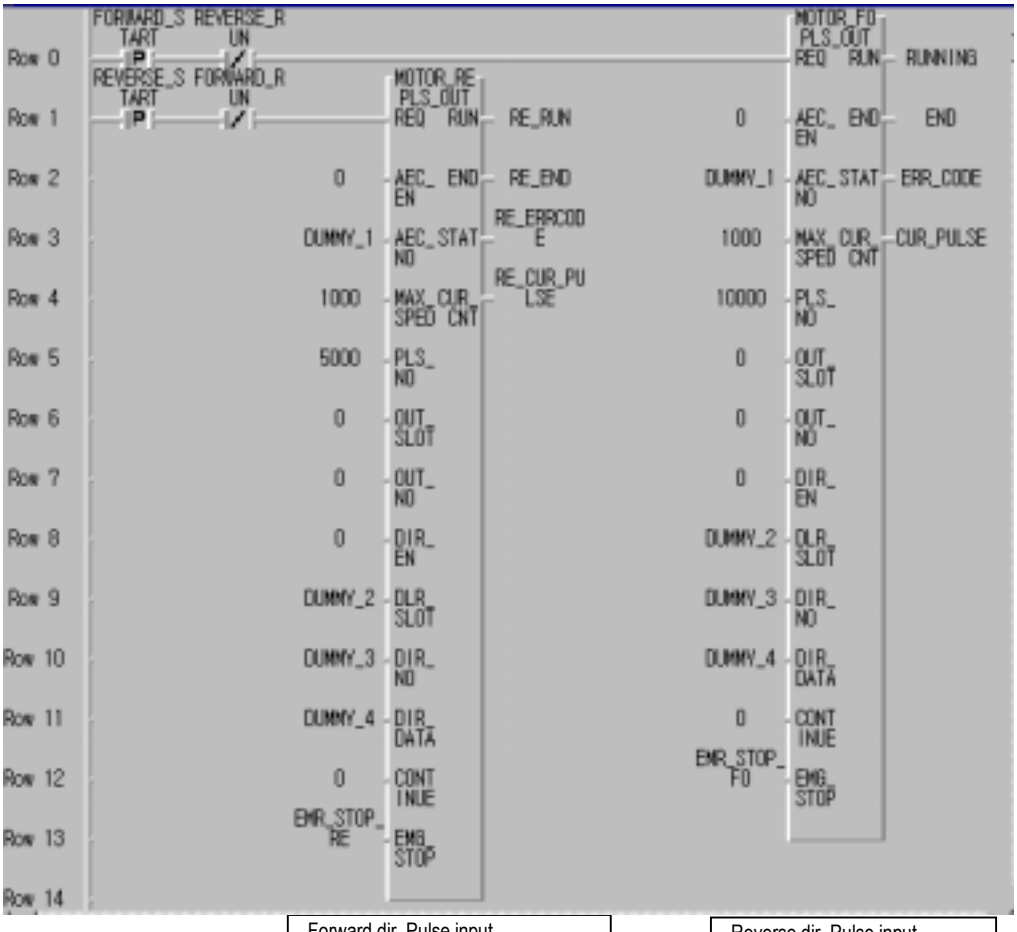
(Example of a program)

(b) Driver gets input forward direction pulse and reverse direction pulse through different contact points.





Velocity Profile



Forward dir. Pulse input

Reverse dir. Pulse input

(Example of a program)

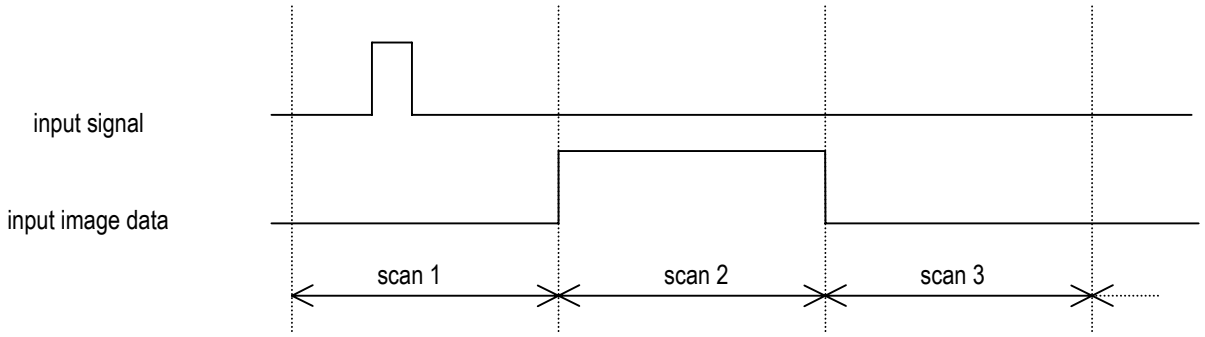
7.1.3. Pulse Catch Function

In the base unit, 8 points of pulse catch input contact points(%IX0.0.0 ~ %IX0.0.7) are internalized. Through using this contact point short pulse signal, short as 0.2ms, can be taken which can not be executed by general digital input.

1) Usage

When narrow width of pulse signal is input, a trouble occurs which can not be executed by general digital input, so the operation does not perform as user's intention. But in this case through pulse catch function even narrow interval of pulse signal as 0.2ms min can be executed.

2) Operating Explanation



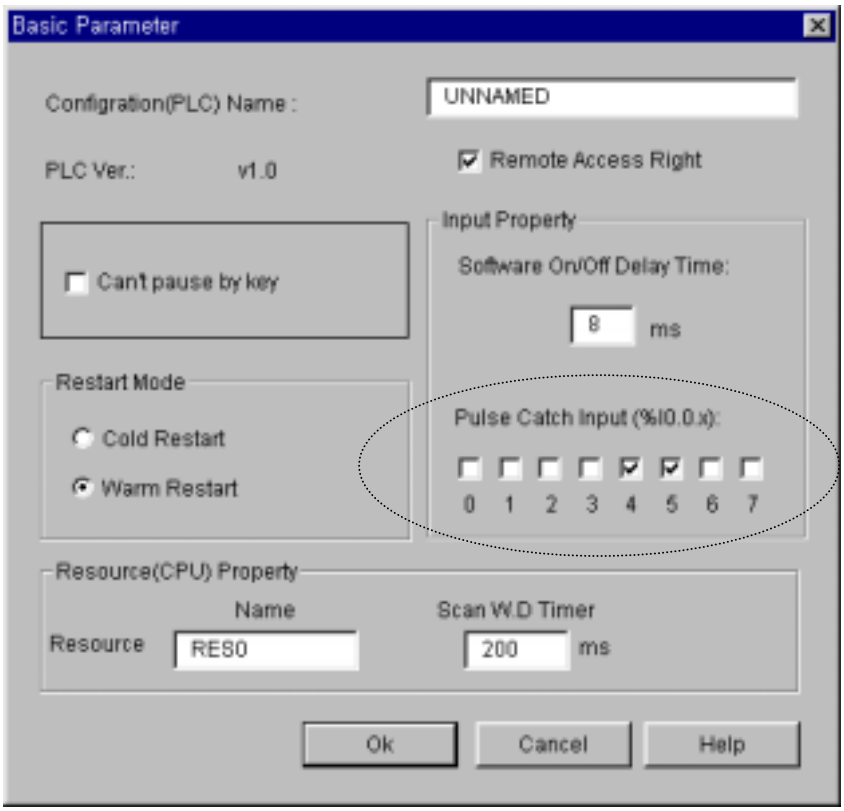
step	executing contents
scan1	CPU senses input when pulse signal, min. 0.2ms, is input, then saves the status.
scan2	used to turn on the region of input image
scan3	used to turn off the region of input image

3) using method

(1) click twice the basic parameter on the project window of GWMIN

(2) Select no. to use for pulse catch input of the basic parameter window.

For details of GMWIN refers to the manual.



Remark

1) 8 points can be used to designate the pulse catch input. The input address is from %IX0.0.0 to %IX0.0.7.
2) General digital input operates if it is not designated as pulse catch input.

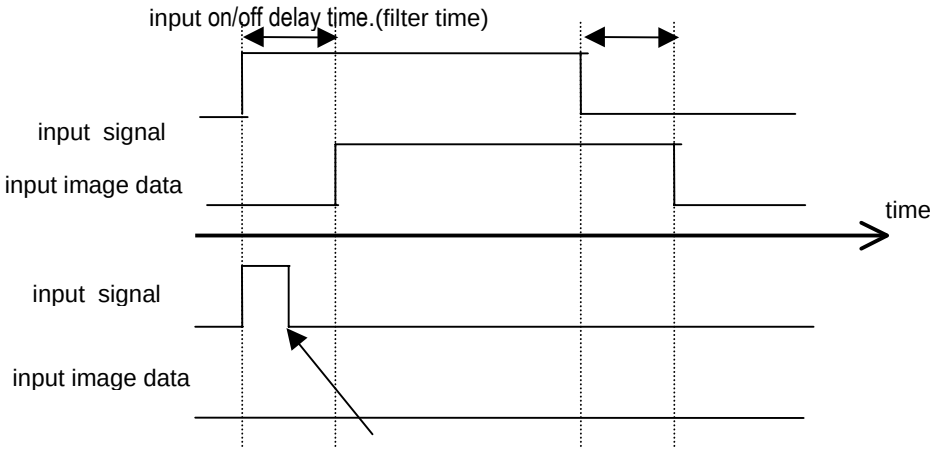
7.1.4. Input Filter Function

External input of GM7 selects input on/off delay time. from the range of 0-15ms of GMWIN. Credibility secured system may be established by adjustment of input correction no. through using environment.

1) Usage

Input signal status affects to the credibility of system in where noise occurs frequently or pulse width of input signal affects as a crucial factor. In this case the user sets up the proper input on/off delay time, then the trouble by miss operation of input signal may be prevented because the signal which is shorter than set up value is not adopted.

2) Operating Explanation



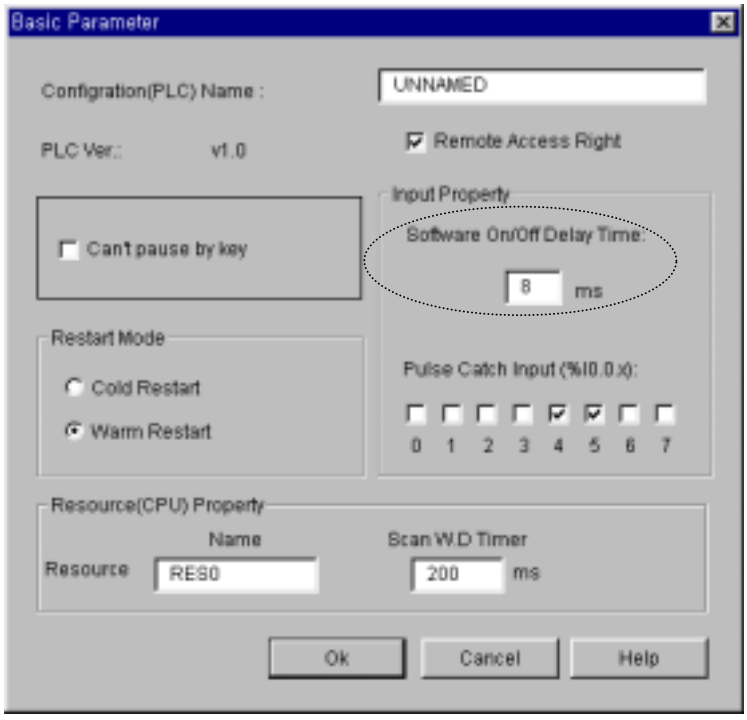
narrower width pulse than input correction no. is not considered as input signal

3) Using method

(1) Click twice the basic parameter on the project window of GMWIN.

(2) The value of filter can be set up as unit of 1ms to the input on/off delay time of the basic parameter window.(Input on/off delay time is set up as default value of 8ms)

(3) Set up input on/off delay time is conformed to all input is used.



7.1.5 PID control function

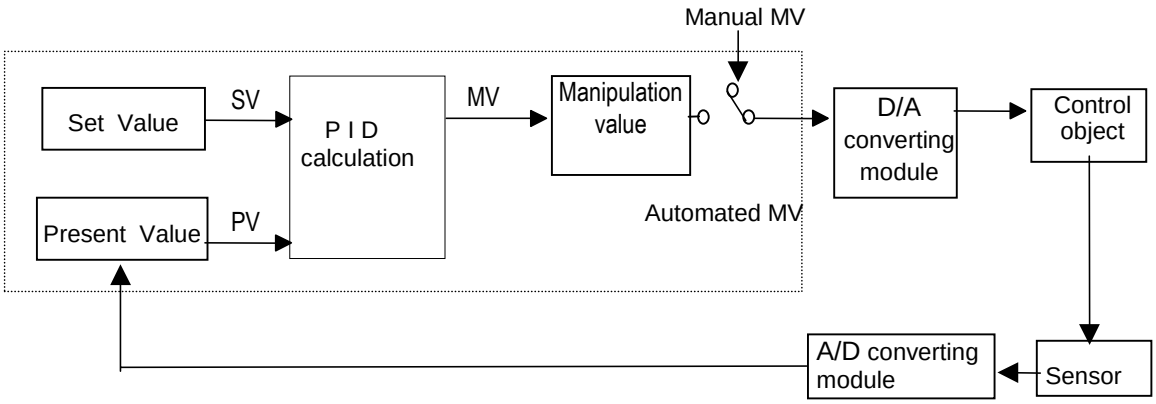
1) Introduction

This chapter will provide information about the built-in PID (Proportional Integral Differential) function of GM7 Basic Unit. The GM7 series does not have separated PID module like GM3 and GM4 series, and the PID function is integrated into the Basic Unit.

The PID control means a control action in order to keep the object at a set value (SV). It compares the SV with a sensor measured value (PV : Present Value) and when a difference between them (E : the deviation) is detected, the controller output the manipulate value (MV) to the actuator to eliminate the difference. The PID control consists of three control actions that are proportional (P), integral (I), and differential (D).

The characteristics of the PID function of GM7 is as following;

- the PID function is integrated into the CPU module. Therefore, all PID control action can be performed with F/B (Function Block) without any separated PID module.
- Forward / reverse operations are available
- P operation, PI operation, PID operation and On/Off operation can be selected easily.
- The manual output (the user-defined forced output) is available.
- By proper parameter setting, it can keep stable operation regardless of external disturbance.
- The operation scan time (the interval that PID controller gets a sampling data from actuator) is changeable for optimizing to the system characteristics.



<Figure 1-1> Flowchart of PID control system

2) Specification

(1) Control operation

a) Proportional operation (P operation)

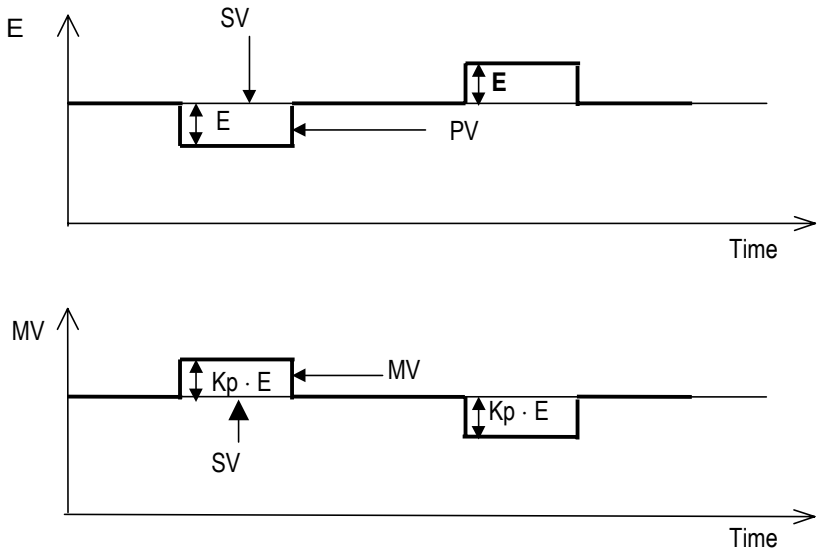
(a) P action means a control action that obtain a manipulate value which is proportional to the deviation (E : the difference between SV and PV)

(b) The deviation (E) is obtained by multiplying a reference value to the actual difference between SV and PV. It prevents the deviation from a sudden change or alteration caused by external disturbance. The formula of deviation is as following;

$$MV = Kp \times [b \times SV - PV]$$

Kp : the proportional constant (gain)
b: reference value
SV: set value
PV: present value

(c) When E happens, MV by P operation is like <Fig 2-1>



<Fig 2-1> MV by P operation

(d) If the Kp is too large, the PV reaches to the SV swiftly, but it may causes a bad effect like oscillations shown in the Fig. 2.2.

(e) If the Kp is too small, oscillation will not occur. However, the PV reaches to the SV slowly and an offset may appear between PV and SV shown in the Fig. 2.3.

(f) The manipulation value (MV) varies from 0 to 4,000. User can define the maximum value of MV (MV_MAX) and minimum value (MV_MIN) within the range 0 ~ 4,000.

(g) When an offset remains after the system is stabilized, the PV can be reached to the SV by adding a certain value. This value is called as bias value, and user can define the bias value

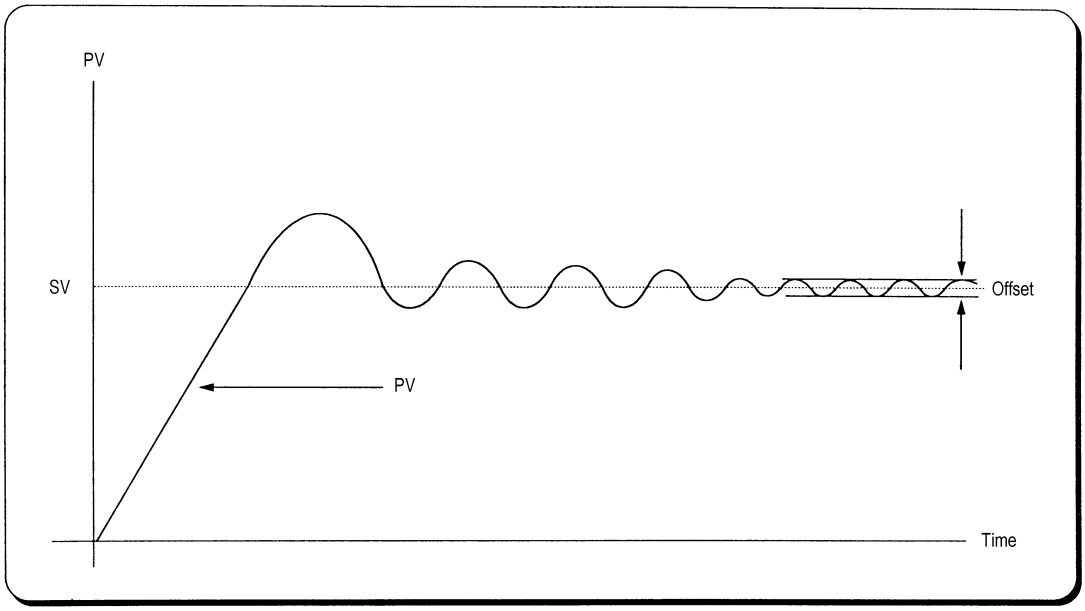


Fig. 2.2 When the proportional constant (Kp) is large

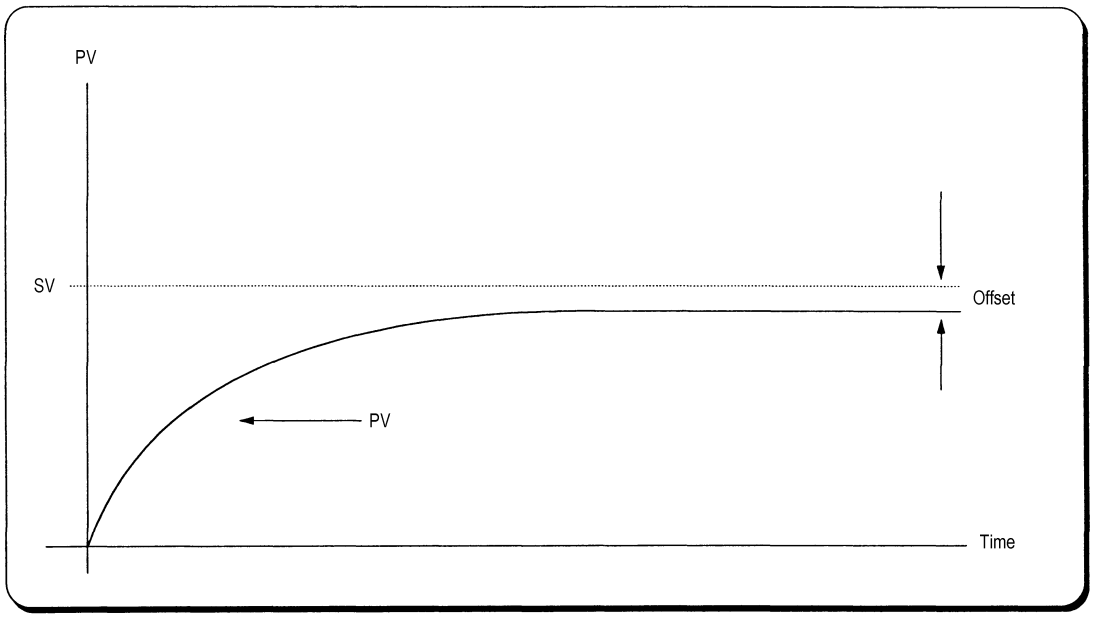


Fig. 2.3 When the proportional constant (Kp) is small

b) Integral operation (I operation)

(a) With integral operation, the manipulate value (MV) is increased or decreased continuously in accordance time in order to eliminate the deviation between the SV and PV. When the deviation is very small, the proportional operation can not produce a proper manipulate value and an offset remains between PV and SV. The integral operation can eliminate the offset value even the deviation is very small.
The period of the time from when the deviation has occurred in I action to when the MV of I action become that of P action is called Integration time and represented as Ti.

(b) Integral action when a constant deviation has occurred is shown as the following Fig. 2.4.

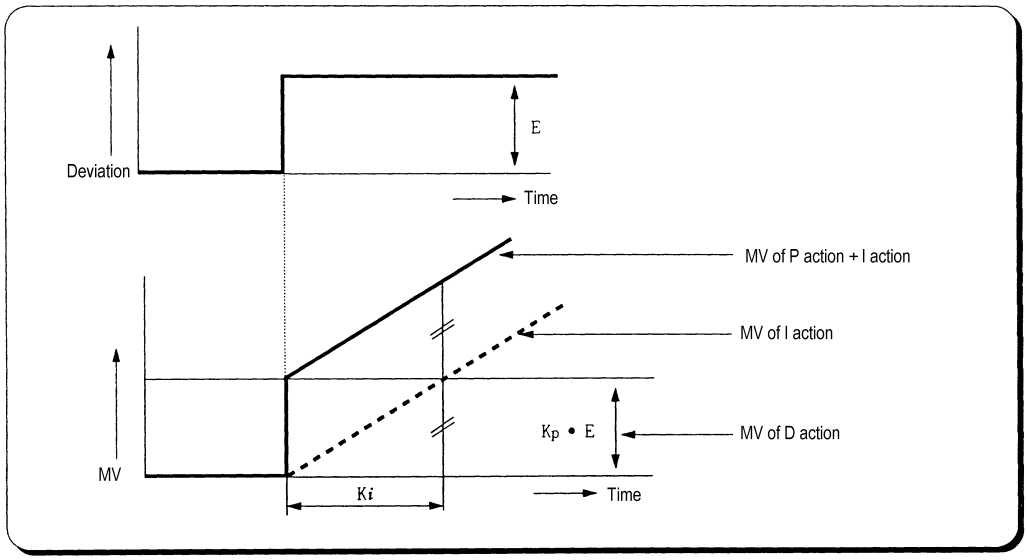


Fig. 2.4 The integral action with constant deviation

(c) The expression of I action is as following;

$$MV = \frac{Kp}{Ti} \int Edt$$

As shown in the expression, Integral action can be made stronger or weaker by adjusting integration time (K*i*) in I action.

That is, the more the integration time (the longer the integration time) as shown in Fig. 2.5, the lesser the quantity added to or subtracted from the MV and the longer the time needed for the PV to reach the SV.

As shown in Fig. 2.6, when the integration time given is short the PV will approach the SV in short time since the quantity added or subtracted become increased. But, If the integration time is too short then oscillations occur, therefore, the proper P and I value is requested.

(d) Integral action is used in either PI action in which P action combines with I action or PID action in which P and D actions combine with I action.

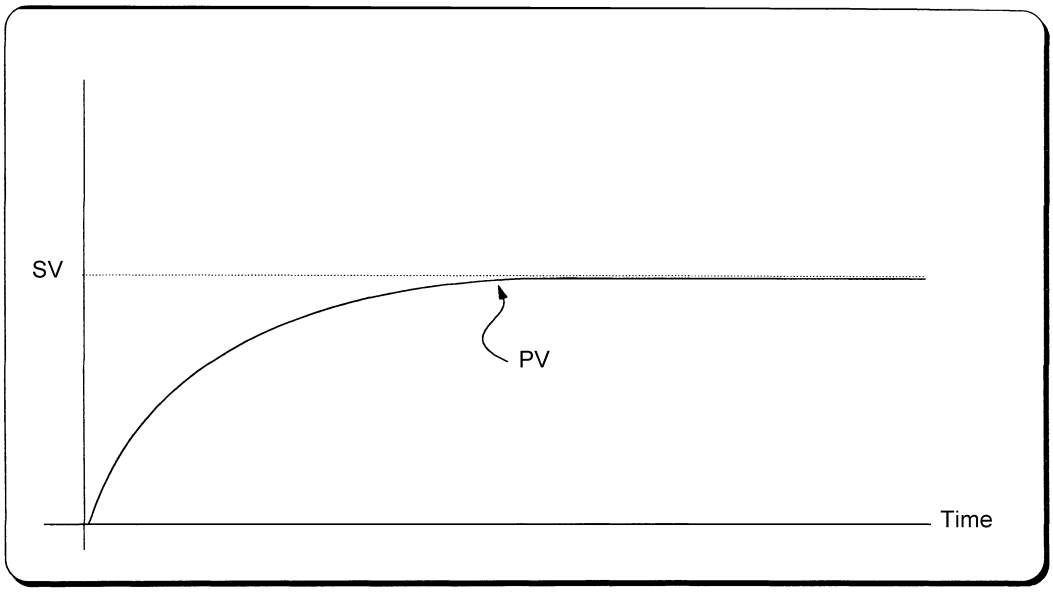


Fig. 2.5 The system response when a long integration time given

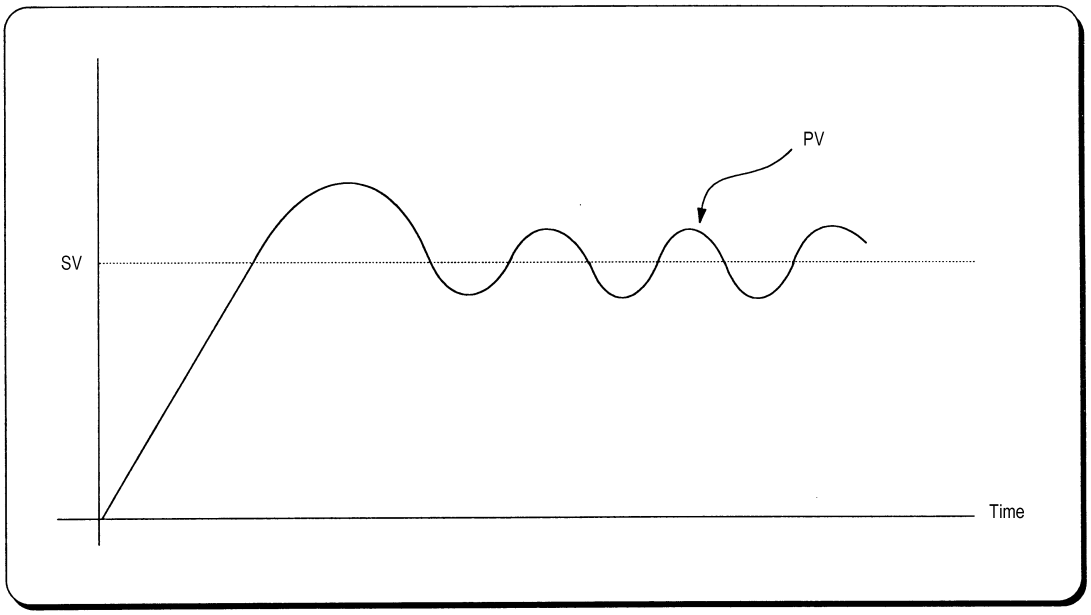


Fig. 2.6 The system response when a short integration time given

c) Derivative operation (D action)

(e) When a deviation occurs due to alteration of SV or external disturbances, D action restrains the changes of the deviation by producing MV which is proportioned with the change velocity (a velocity whose deviation changes at every constant interval) in order to eliminate the deviation.

- D action gives quick response to control action and has an effect to reduce swiftly the deviation by applying a large control action (in the direction that the deviation will be eliminated) at the earlier time that the deviation occurs.
- D action can prevent the large changes of control object due to external conditions.

(f) The period of time from when the deviation has occurred to when the MV of D action become the MV of P action is called derivative time and represented as Kd.

(g) The D action when a constant deviation occurred is shown as Fig. 2.7.

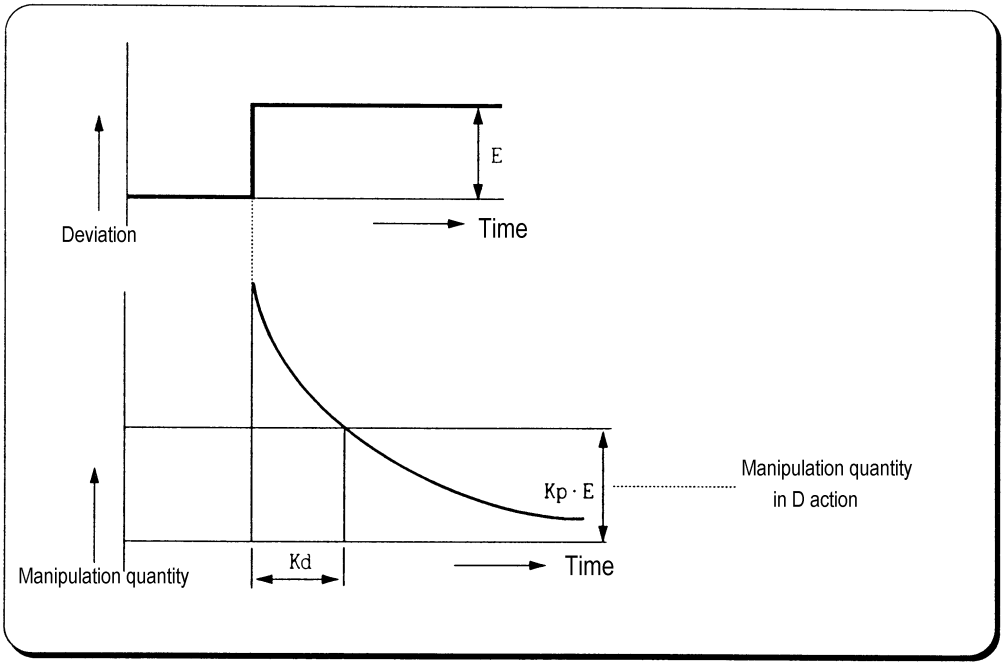


Fig. 2-7 Derivative action with a constant deviation

(h) The expression of D action is as following;

$$MV = Kp \times Td \frac{dE}{dt}$$

(i) Derivative action is used only in PID action in which P and I actions combine with D action.

d) PID action

(a) PID action controls the control object with the manipulation quantity produced by (P+I+D) action

(b) PID action when a given deviation has occurred is shown as the following Fig. 2.8.

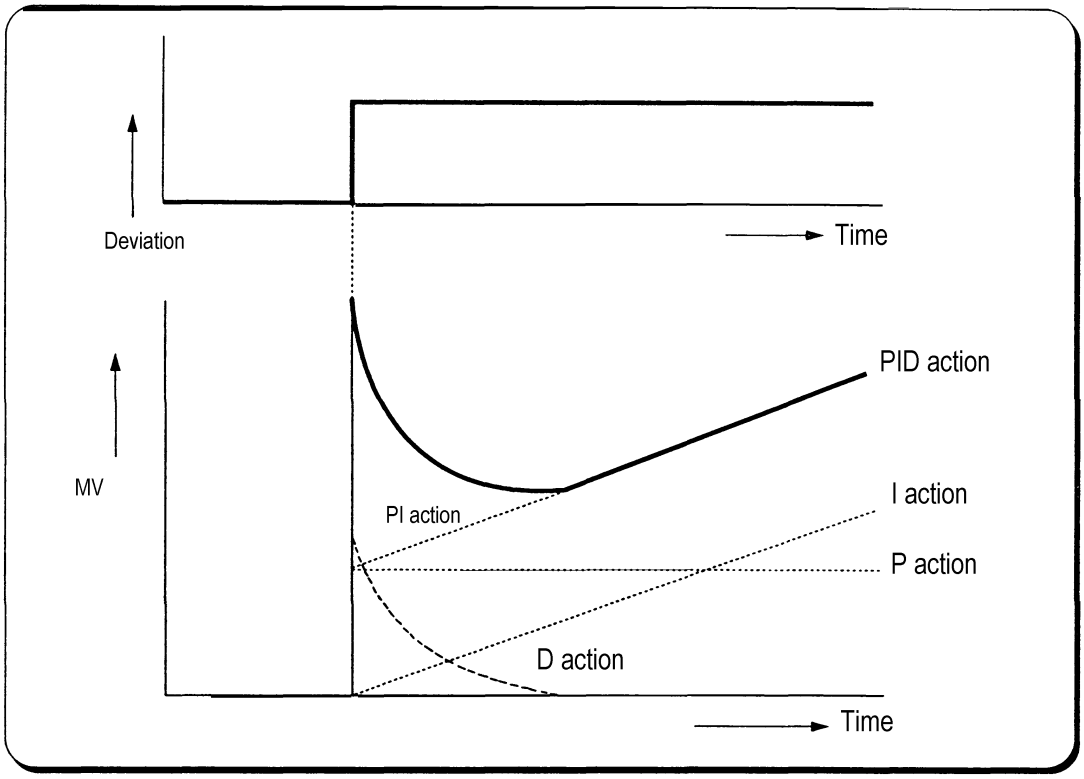


Fig. 2-8 PID action with a constant deviation

e) Forward / Reverse action

(c) PID control has two kind of action, forward action and reverse action. The forward action makes the PV reaches to SV by outputting a positive MV when the PV is less than SV.

(d) A diagram in which forward and reverse actions are drawn using MV, PV and SV is shown as Fig. 2.9.

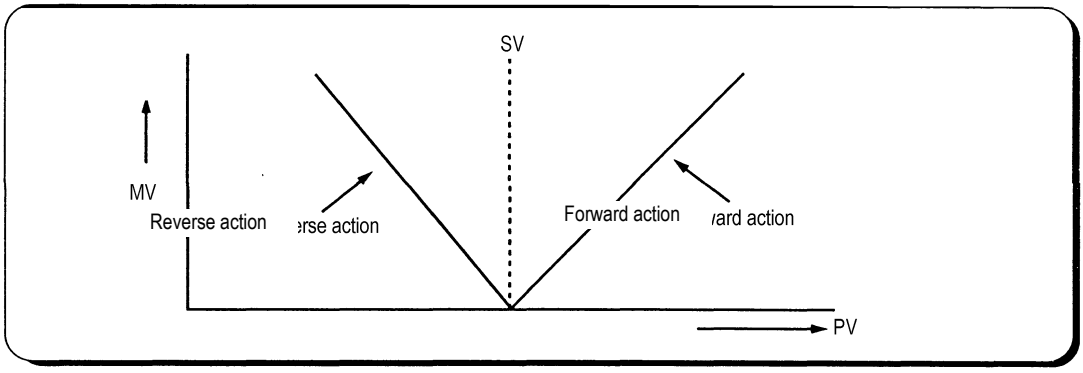


Fig. 2-9 MV of forward / reverse action

(e) Fig 2.10 shows examples of process control by forward and reverse actions, respectively.

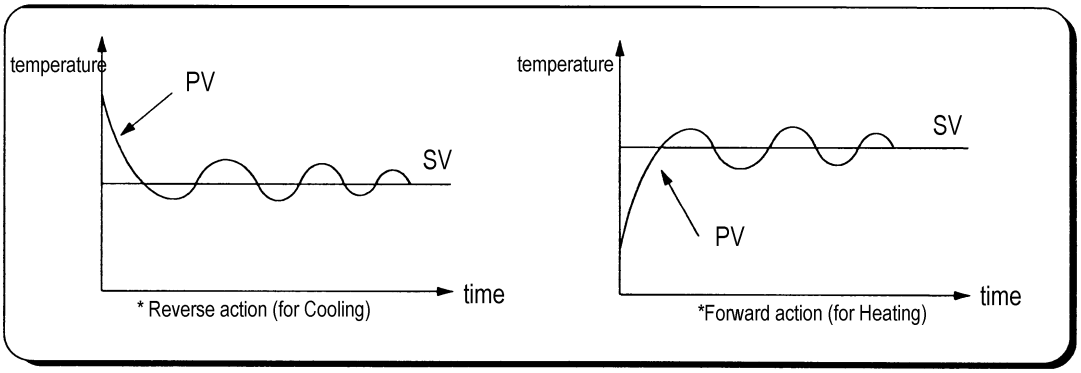


Fig. 2-10 PV of forward / reverse action

f) Reference value

In general feedback control system shown as the Figure 2-10, the deviation value is obtained by the difference of PV and SV. P, I, and D operations are performed based on this deviation value. However, each of P, I, and D operations use different deviation values according to the characteristics of each control actions. The expression of PID control is as following;

$$MV = K\left[Ep + \frac{1}{Ti}\int_0^t Ei(s)ds + Td\frac{dEd}{dt} \right]$$

MV : Manipulate value
K: Proportional gain
Ti: Integral time
Td: Derivative time
Ep: Deviation value for proportional action
Ei: Deviation value for integral action
Ed: Deviation value for derivative action

The deviation values of P, I, and D action is described as following equations;

$$Ep = b \times SV - PV$$
$$Ei = SV - PV$$
$$Ed = -PV$$

The b of the first equation is called as reference value. It can be varied according to the load disturbance of measurement noise.

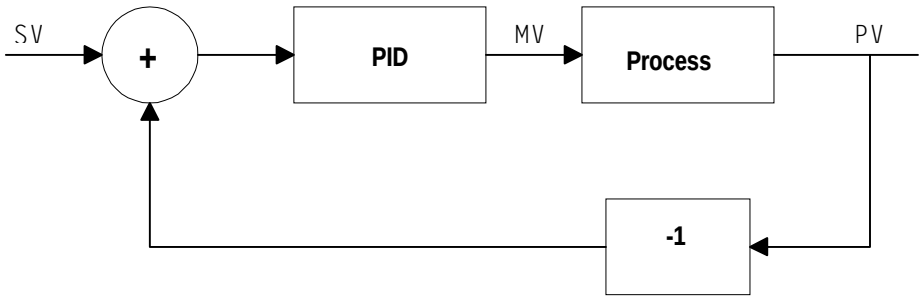


Fig. 2-11 Diagram of simple feedback system

The figure 2.11 shows the variation of PV according to the several different reference values (b). As shown in the Fig. 2.11, the small reference value produces small deviation value, and it makes the control system response be slow.
In general, control system is required to be adaptable to various external / internal changes. Especially, it should shows a stable transient response with the sudden change of the SV to be robust to load disturbances and/or measurement noise.

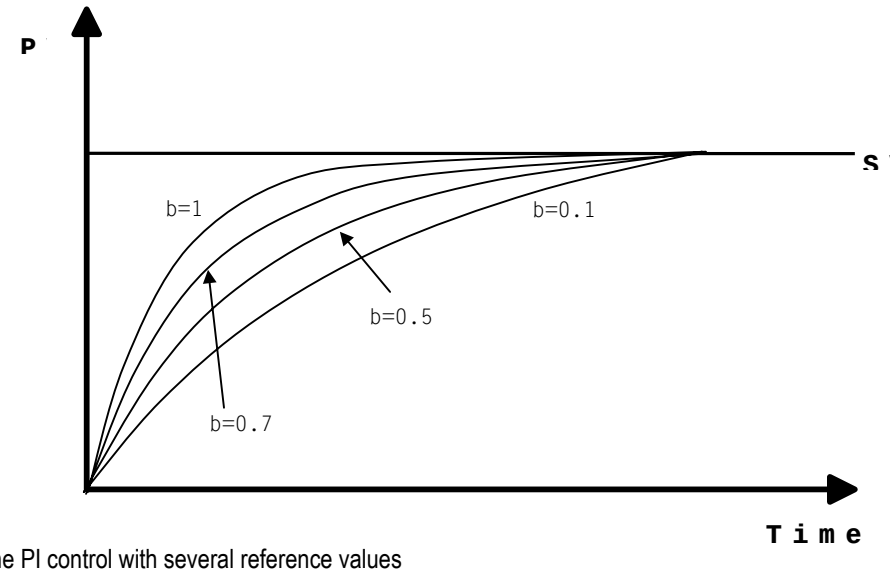
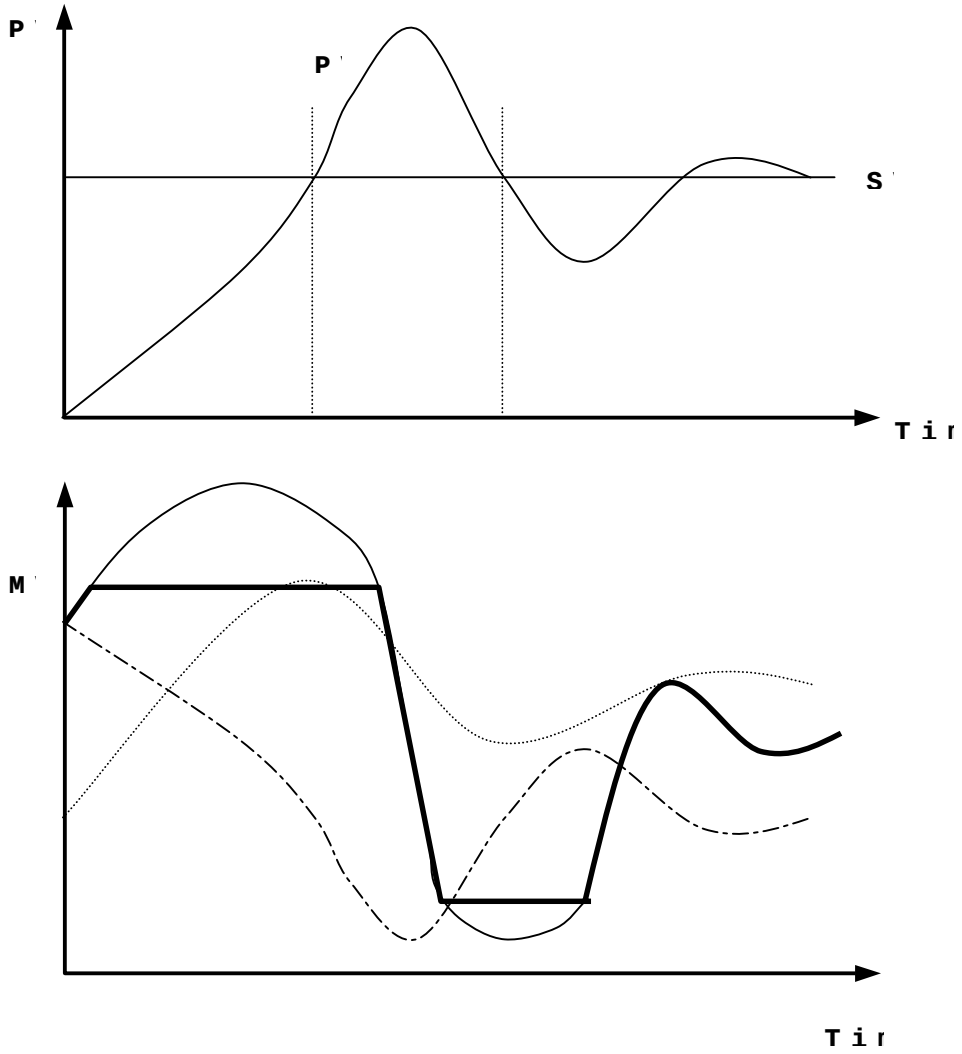


Figure 2-11 The PI control with several reference values

g) Integral windup

All devices to be controlled, actuator, has limitation of operation. The motor has speed limit, the valve can not flow over the maximum value. When the control system has wide PV range, the PV can be over the maximum output value of actuator. At this time, the actuator keeps the maximum output regardless the change of PV while the PV is over the maximum output value of actuator. It can shorten the lifetime of actuator.
When the I control action is used, the deviation term is integrated continuously. It makes the output of I control action very large, especially when the response characteristic of system is slow.
This situation that the output of actuator is saturated, is called as 'windup'. It takes a long time that the actuator returns to normal operating state after the windup was occurred.
The Fig. 2-12 shows the PV and MV of PI control system when the windup occurs. As shown as the Fig. 2-12, the actuator is saturated because of the large initial deviation. The integral term increase until the PV reaches to the SV (deviation = 0), and then start to decrease while the PV is larger than SV (deviation < 0). However, the MV keeps the saturated status until the integral term is small enough to cancel the windup of actuator. As the result of the windup, the actuator will output positive value for a while after the PV reached to the SV, and the system show a large overshoot. A large initial deviation, load disturbance, or mis-operation of devices can cause windup of actuator.



There are several methods to avoid the windup of actuator. The most popular two methods are adding another feedback system to actuator, and using the model of actuator. The Fig. 2-13 shows the block diagram of the anti-windup control system using the actuator model.

As shown in the Fig. 2-13, the anti-windup system feedback the multiplication of gain (1/Tt) and Es to the input of integral term. The Es is obtained as the difference value between actuator output (U) and manipulation value of PID controller (MV). The Tt of the feedback gain is tracking time constant, and it is in inverse proportion with the resetting speed of integral term. Smaller Tt will cancel the windup of actuator faster, but too small Tt can cause anti-windup operation in derivative operation. The Fig. 2-14 shows several Tt value and PV in the PI control system.

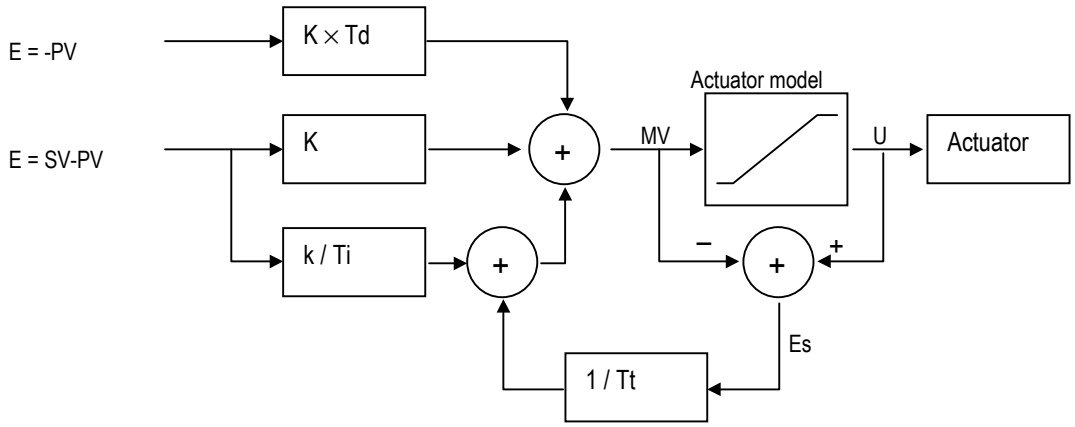


Fig. 2-13 The block diagram of anti-windup control system

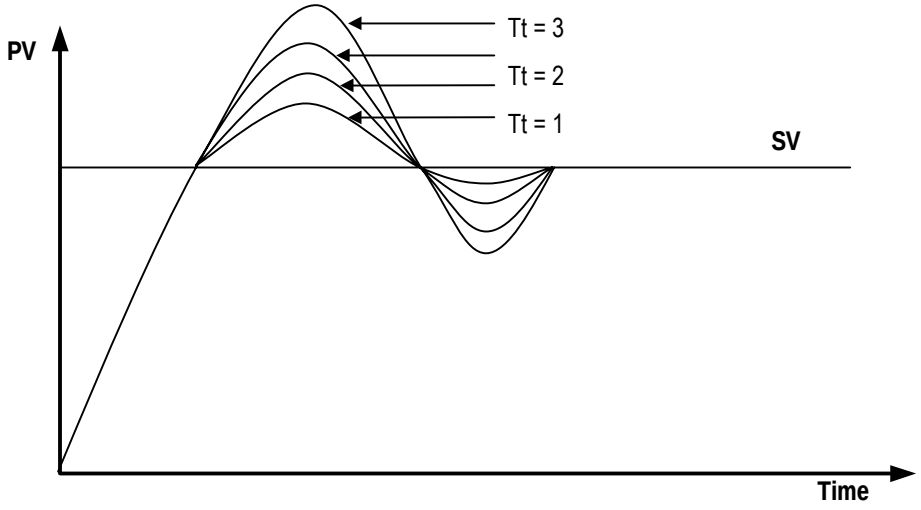


Fig. 2-14 The PV output characteristics with different Tt values.

(2) Realization of PID control on the PLC

In this chapter, it will described that how to get the digitized formula of the P, I, and D terms. Then, the pseudo code of PID control will be shown.

a) P control

The digitized formula of P control is as following;

$$P(n) = K\left[b \times SV(n) - PV(n)\right]$$ n : sampling number

K : proportional gain constant
b : reference value
SV : set value
PV : present value

b)I control

The continuous formula of I control is as following;

$$I(t) = \frac{K}{Ti}\int_0^t e(s)ds$$ I(t) : integral term

K : proportional gain constant
Ti : integral time
e(s) : deviation value
By deviation about t, we can obtain;

$$\frac{dI}{dt} = \frac{K}{Ti} e$$ e = (SV – PV) : deviation value

The digitized formula is as following;

$$\frac{I(n+1) - I(n)}{h} = \frac{K}{Ti} e(n)$$ h : sampling period

$$I(n+1) = I(n) + \frac{Kh}{Ti} e(n)$$

c) D control

The continuous formula of derivative term is as following;

$$\frac{Td}{N} \times \frac{d}{dt} D + D = -KTd \frac{dy}{dt}$$

N : high frequency noise depression ration
y : the object to be controlled (PV)

The digitized formula is as following (Use Tustin approximation method)

$$D(n) = \frac{2Td - hN}{2Td + hN} D(n-1) - \frac{2KTdN}{2Td + hN} \left[y(n) - y(n-1) \right]$$

d) Pseudo code of PID control

The pseudo code of PID control is as following;

- Step 1 : Get constants that are used for PID operation

$Bi = K \times \frac{h}{Ti}$: integral gain

$Ad = \frac{(2 \times Td - N \times h)}{(2 \times Td + N \times h)}$: derivation gain

$Bd = \frac{(2 \times K \times N \times Td)}{(2 \times Td + N \times h)}$

$A0 = \frac{h}{Tt}$: anti-windup gain

- Step 2 : Read SV and PV value
  PV = adin(ch1)
- Step 3: Calculate the proportional term.
  P = K × (b × SV – PV)
- Step 4 : Update the derivative term. (initial value of D = 0)
  D = As × D – Bd × (PV – PV_old)
- Step 5 : Calculate the MV. (initial value of I = 0)
  MV = P + I + D
- Step 6 : Check the actuator is saturated or not.
  U = sat(MV, U_low, U_high)
- Step 7 : Output the MV value to the D/A module
- Step 8 : Update the integral term.
  I = I + bi × (SV – PV) + A0 × (U – MV)
- Step 9 : Update the PV_old value.
  PV_old = PV

3) Function block

For the PID operation of GM7, following 2 function blocks are included in the GMWIN software. (version 3.3 or later)

No	Name	Description
1	PID7CAL	Perform the PID operation
2	PID7AT	Perform the auto tuning operation

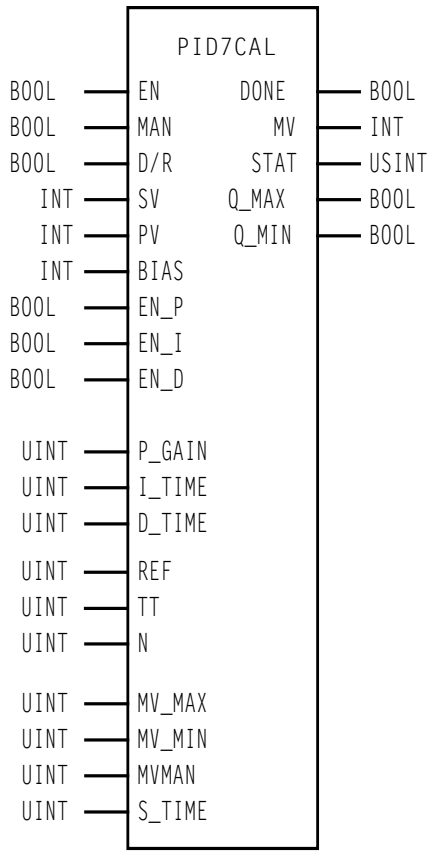
Remarks

1.GM7 PID function blocks do not support array type.

2.Refer the GMWIN manual for the registration and running of function block.

(1) The function block for PID operation (PID7CAL)

a) Description of F/B

Function block	Description
	**Input** EN : enable signal of the PID6CAL F/B MAN : manual operation mode (0 : auto, 1 : manual) D / R : select direction of operation (0 : forward, 1 : reverse) SV(*1) : set value data input (input range : 0 ~ 4000) PV(*1) : present value data input BIAS (*2): feed forward or offset value input for disturbance compensation (input range : 0 ~ 4000) EN_P(*3) : enable signal of proportional control (0 : disable, 1 : enable) EN_I(*3) : enable signal of integral control (0 : disable, 1 : enable) EN_D(*3) : enable signal of derivative control (0 : disable, 1 : enable) P_GAIN(*4) : the proportional gain constant (range : 0.01 ~ 100.00) I_TIME(*5) : the integration time (range : 0.0 ~ 2000.0) D_TIME(*5) : the deviation time (range : 0.0 ~ 2000.0) MV_MAX : the maximum value of MV (range : 0 ~ 4000) MV_MIN : the minimum value of MV (range : 0 ~ 4000) MVMAN : the input data of manual operation mode (range : 0 ~ 4000) S_TIME(*6) : operation scan time (range : 0.1 ~ 10) REF(*7) : the reference value (range : 0.1 ~ 1) TT(*8) : tracking time constant (range : 0.01 ~ 10.00) N(*9) : high frequency noise depression ratio (range : 1 ~ 10) **Output** DONE : completion flag of PID operation MV : output manipulation value (range : 0 ~ 4000) STAT : error code output Q_MAX : shows MV is limited with maximum value Q_MIN : shows MV is limited with minimum value

(*1) SV (setting value : the designated value) and PV (process value : present value) of GM7 PID operation have the range 0 ~ 4000. The range is set with the consideration of the resolution of A/D and D/A module of GM7 series (12 bits) and offset value.
(*2) The BIAS data is used for the compensation of offset in the proportional control.
(*3) In GM7, only the following 4 operation modes are available. Other operation modes, such as PD or I, are not permitted.

No.	EN_P	EN_I	EN_D	Operation
1	1 (enable)	0 (disable)	0 (disable)	P operation
2	1 (enable)	1 (enable)	0 (disable)	PI operation
3	1 (enable)	1 (enable)	1 (enable)	PID operation
4	0 (disable)	0 (disable)	0 (disable)	On/Off operation

(*4) The GM7 can handle only integer, not the floating point type. Therefore, to enhance the accuracy of PID operation, the PID6CAL function block is designed to input the P_GAIN data as the 100 times scaled up. For example, if the designated P_GAIN is 98, actual input data of P_GAIN should be 9800. If the designated P_GAIN is 10.99, input 1099 to the P_GAIN.
(*5) I_TIME and D_TIME are 10 times scaled up. For example, input 18894 if the designated I_TIME value is 1889.4. The range of actual input is 0 ~ 20000.
(*6) S_TIME is the period of reading data (sampling), and also 10 times scaled up. Generally, it should be synchronized with external trigger input (EN input of function block) to perform proper PID operation. The range of sampling time is 0.1 ~ 10 seconds, and actual input range is 0 ~ 100.
(*7) REF may be useful parameter according to the control system type, especially velocity, pressure, or flux control system. The REF input is also 10 times scaled up, and the actual range is 0 ~ 10.
(*8) TT (tracking time constant) parameter is used to cancel anti-windup operation. The range of TT is 0.01 ~ 10 and the actual input range that are 100 times scaled up is 0 ~ 1000.
(*9) N (high frequency noise depression ratio) parameter is used for derivative control operation, and shows the ratio of high frequency noise depression. If there is a lot of high frequency noise in the control system, select the N value as higher value. Otherwise, leave the N parameter as 1. The range of N is 0 ~ 10 and it is not scaled up, so input the designated value directly.

b) The error code of PID7CAL F/B

The following table shows error codes and descriptions of PID7CAL function block.

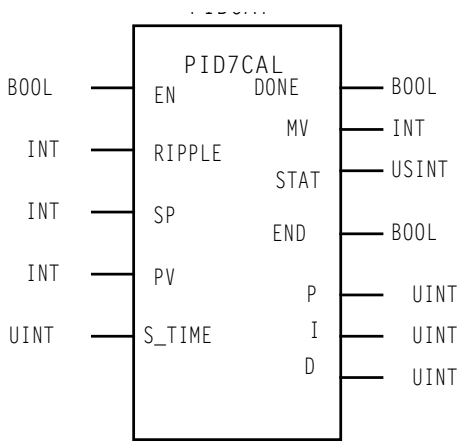
Error code (STAT output)	Description	Countermeasure
0	Normal operation	
1	SV is out of range	Change the SV within 0 ~ 4000
2	MVMAN is out of range	Change the MVMAN within 0 ~ 4000
3	P_GAIN is out of range	Change the P_GAIN within 0 ~ 10000
4	I_TIME is out of range	Change the I_TIME within 0 ~ 20000
5	D_TIME is out of range	Change the D_TIME within 0 ~ 20000
6	S_TIME is out of range	Change the S_TIME within 0 ~ 100
7	REF is out of range	Change the REF within 0 ~ 10
8	TT is out of range	Change the TT within 0 ~ 1000
9	N is out of range	Change the N within 0 ~ 1000
10	EN_I and/or EN_D is set as 1 when EN_P is 0	Only P, PI, and PID controls are available. Please change the setting of EN_P, EN_I, and EN_D.

Remark

1. Please be careful to input 100 times scaled up values for P_GAIN and TT.
2. I_TIME, D_TIME, S_TIME, and REF are 10 times scaled up, not 100 times.

(2) Auto tuning function block (PID7AT)

a) Description of F/B

Function block	Description
	**Input** EN : enable input of function block SV(*1) : set value (goal value) data input (range : 0 ~ 4000) PV (*1): present value input (range : 0 ~ 4000) S_TIME(*2) : scan time input (sampling interval) (range : 0 ~ 100) RIPPLE(*3) : select the wave form to be used for auto tuning operation. Select 1 in general case. **Output** DONE : Turn on whenever the auto tuning operation is completed. END : Turns on when the F/B operation is completed with no error, and keep the status until next F/B execution STAT : shows the error code MV : the manipulated value of current loop on which the auto tuning operation is performed. (range : 0 ~ 4000) P : the proportional gain constant obtained by auto tuning operation. (range : 0.01 ~ 100.00) I : the integral time constant obtained by auto tuning operation. D : the derivative time constant obtained by auto tuning operation

*1)SV (setting value : the designated value) and PV (process value : present value) of GM7 PID operation have the range 0 ~ 4000. The range is set with the consideration of the resolution of A/D and D/A module of GM7 series (12 bits) and offset value. When setting the SV or PV, please be careful convert the analog value of control object (temperature, velocity, etc.) to digital value that are the output of A/D convert module. For example, assume that PID control is used for temperature control with Pt100 (operation range : 0 °C ~ 250 °C), and the goal value is 100 °C. The equivalent digital output of A/D module (voltage output range : 1 ~ 5V) is 1600 if the A/D module outputs 0 (1V) with 0 °C, and 4000(5V) with 250 °C. Therefore, the input of SV should be 1600, not 2.

*2)S_TIME is the period of reading data (sampling), and 10 times scaled up for more precious operation. Generally, it should be synchronized with external trigger input (EN input of function block) to perform proper PID operation. The range of sampling time is 0.1 ~ 10 seconds, and actual input range is 0 ~ 100.

*3)The GM7 perform auto-tuning operation based on the frequency response method. PID parameters are obtained by On/Off operation during 1 cycle of PV variation. The RIPPLE parameter shows at which cycle the CPU module will perform auto-tuning operation. If 0 is selected, the CPU will get PID parameters during the first cycle of PV variation. If 1 is selected, the second cycle will be used. (refer Fig. 3-1 for detailed information) Other choice of RIPPLE parameter is not allowed. In general case, select 1 for proper auto-tuning operation. The On/Off operation will be occur at the 80% of PV value.

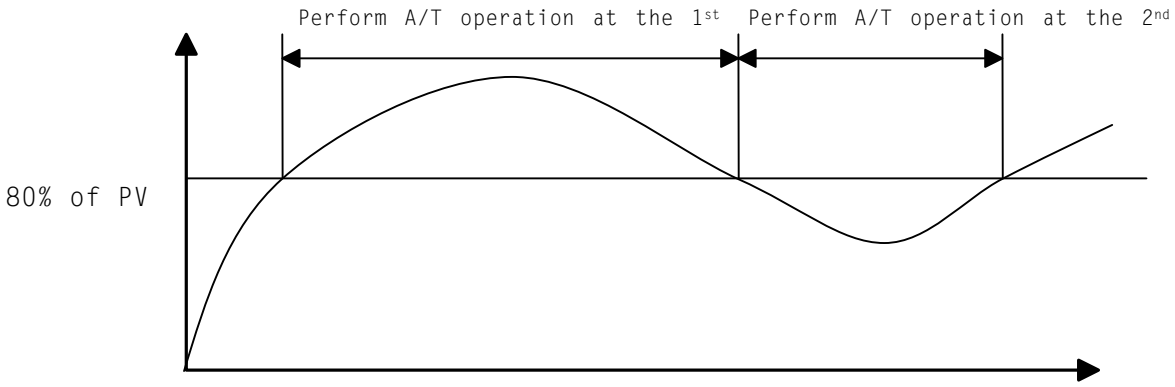


Fig.3-1 The ripple parameter

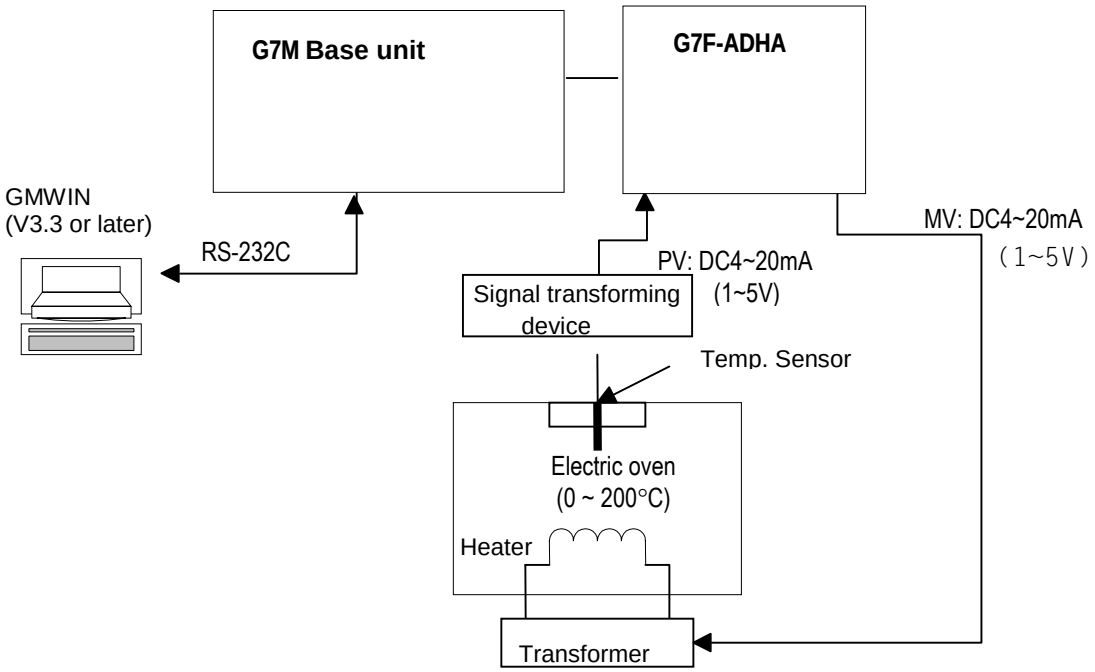
a) Error codes of auto-tuning function block (PID7AT)

The following table shows error codes and descriptions of PID7AT function block.

Error code (STAT output)	Description	Countermeasure
0	Normal operation	
1	SV is out of range	Change the SV within 0 ~ 4000
2	PV is out of range	It may caused by fault of A/D module. Check the A/D module.
3	S_TIME is out of range	Change the S_TIME within 0 ~ 100
32	Ripple is out of range	Change the Ripple to 0 Or 1.

4) Program Example

(1) System configuration



(2) Initial setting

a) PID operation parameters

(a) Auto / Manual operation setting: Auto
(b) Forward / Reverse operation: Forward
(c) SV setting: 1600 (100°C)
(d) BIAS setting: 0 (If only P control is used, input proper value other 0)
(e) EN_P, EN_I, EN_D setting: EN_P=1, EN_I=1, EN_D=1 (PID operation)
(f) REF, TT, N: REF=10, TT=5-, N=1
(g) MV_MAX, MV_MIN, MVMAN: MV_MAX=4000, MC_MIN=0, MAMAN=2000
(h) S_TIME: S_TIME=100 (sampling time = 10 seconds)

b) Auto-tuning parameters

(a) PV setting: 1600 (100°C)
(b) S_TIME: S_TIME=100 (sampling time = 10 seconds)

c) A/D module setting

(c) Channel setting: use channel 0
(d) Output data type: – 48 ~ 4047
(e) Input processing: Sampling

d) D/A module setting

(f) Channel setting: use channel 0

(3) Program Explanation

a) Use only PID operation (without A/T function)

(g) Convert the measured temperature (0 ~ 250°C) to current signal (4 ~ 20mA), and input the current signal to the channel 0 of A/D module. Then, the A/D module converts the analog signal to digital value (0 ~ 4000)
(h) PID6CAL function block will calculate manipulate value (MV : 0 ~ 4000) based on PID parameter settings (P_GAIN, I_TIME, D_TIME, etc.) and PV from A/D module. Then, the calculated MV is output to the channel 0 of D/A module.
(i) D/A module will convert the MV (0 ~ 4000) to analog signal (4 ~ 20mA) and output to the actuator (power converter).

b) Use PID operation with A/T function

(j) Convert the measured temperature (0 ~ 250°C) to current signal (4 ~ 20mA), and input the current signal to the channel 0 of A/D module. Then, the A/D module converts the analog signal to digital value (0 ~ 4000)

(k) A/T function block will calculate manipulate value (MV : 0 ~ 4000) based on the SV and PV from A/D module. Simultaneously, the A/T module will calculate P,I and D parameters.

(l) The END output of A/T module will be 1 when the A/T operation is completed. Then, PID module will start operation with PID parameters that are calculated by A/T module.

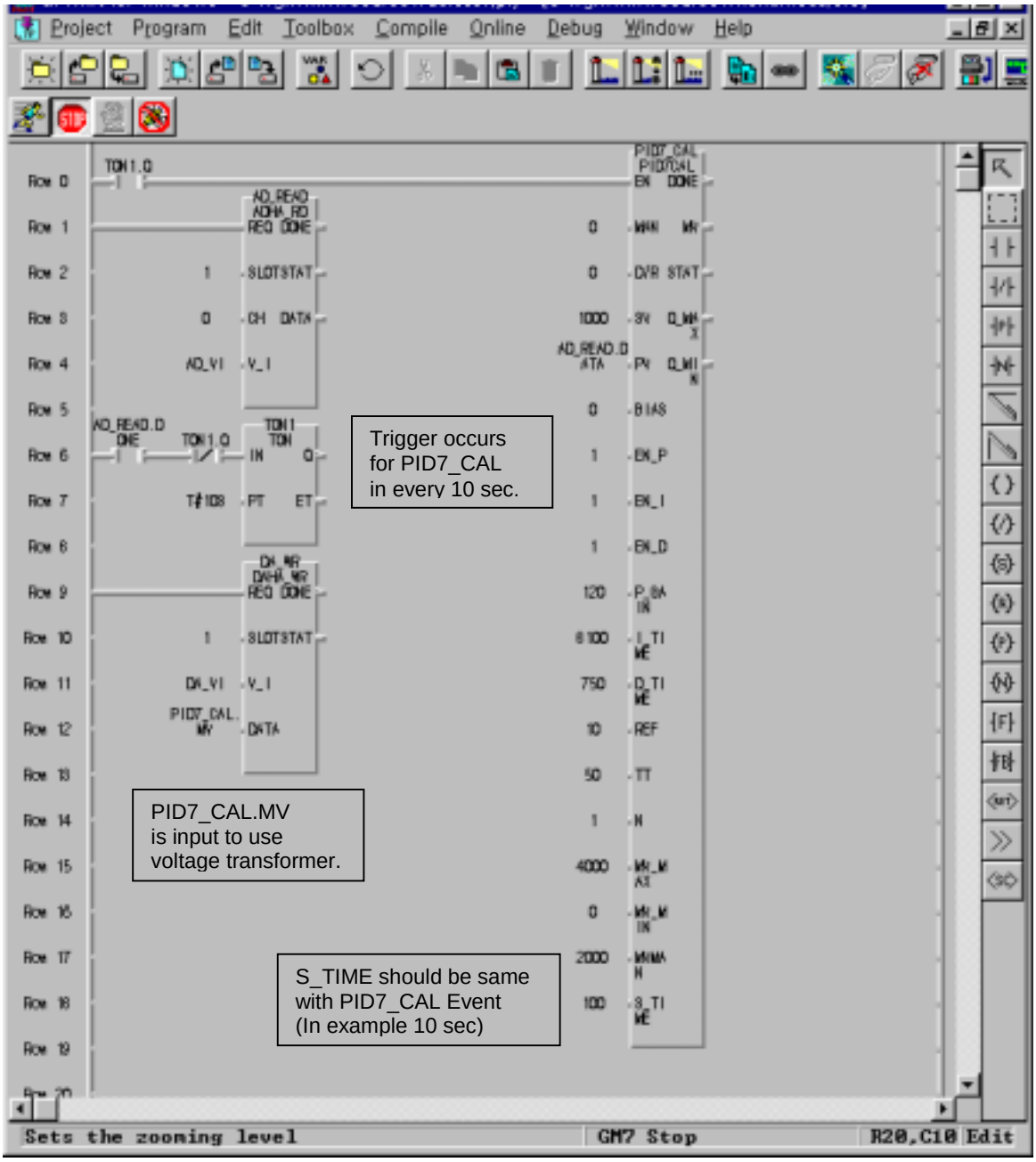
(m) D/A module will convert the MV (0 ~ 4000) to analog signal (4 ~ 20mA) and output to the actuator (power converter).

Remark

G7F-ADHA module is supplied 2channels for A/D exchange and 1channel for D/A exchange module.

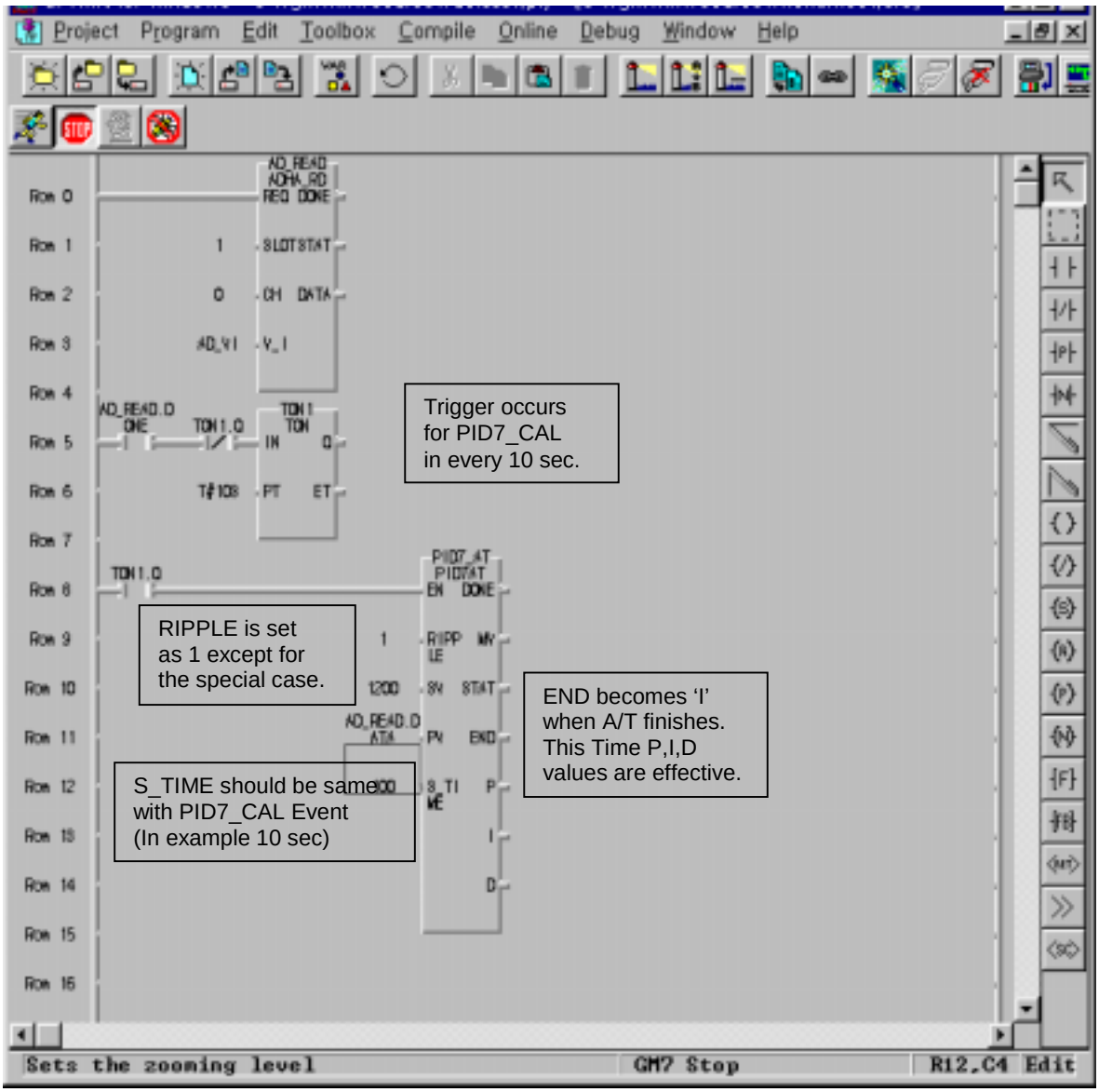
(4) Program

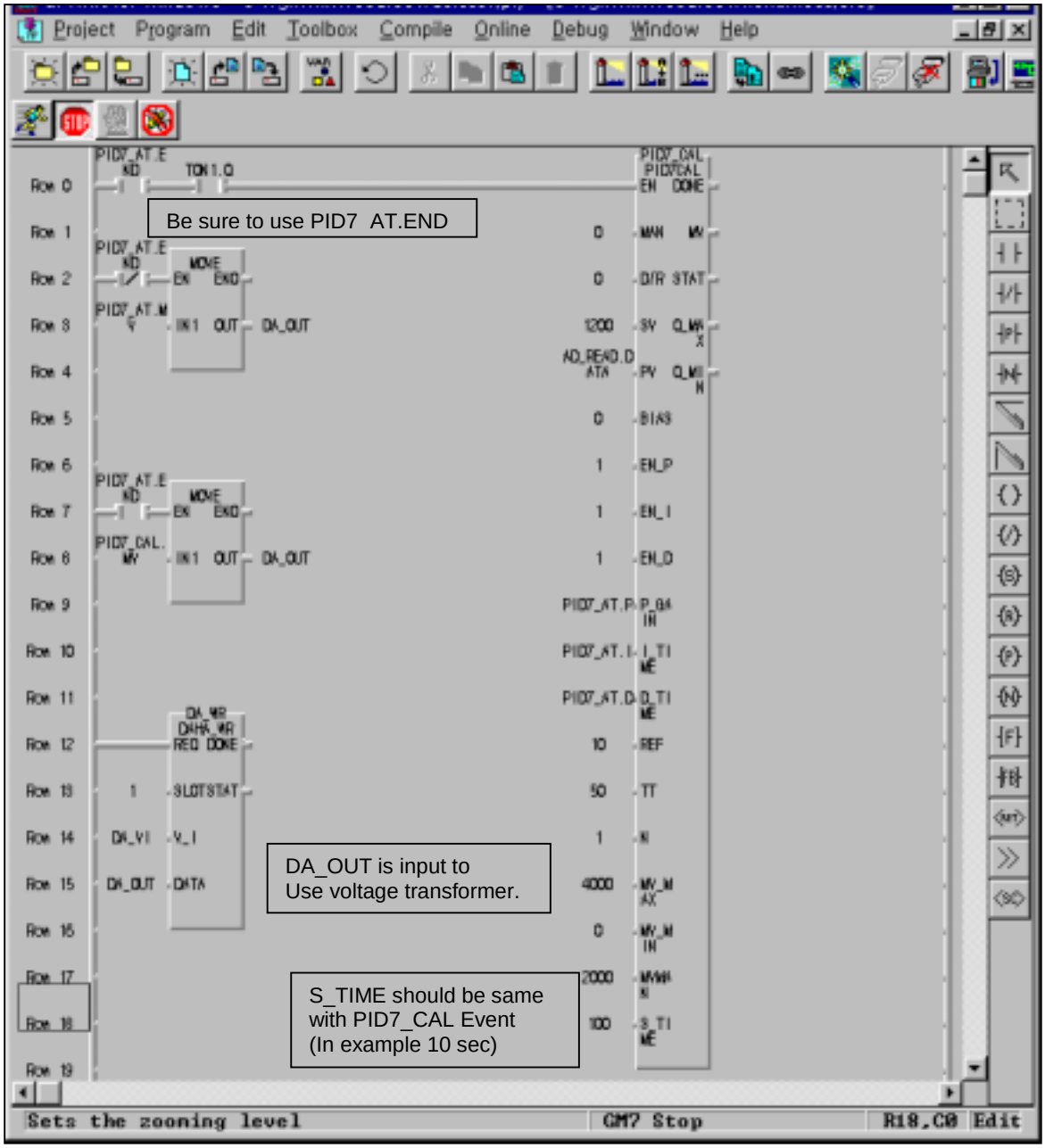
a) In case of using PID function only.


Project Program Edit Toolbox Compile Online Debug Window Help
Trigger occurs for PID7_CAL in every 10 sec.
PID7_CAL.MV is input to use voltage transformer.
S_TIME should be same with PID7_CAL Event (In example 10 sec)
Sets the zooming level
GM7 Stop
R20,C10 Edit

b) In case of using combined function of PID operation and Auto tuning.

This program is an example of PID operation performing with computed P,I,D values by the auto tuning performing. It is performed in 80% of auto tuning SV, PID process is performed from 80% of SV.


Project Program Edit Toolbox Compile Online Debug Window Help
Trigger occurs for PID7_CAL in every 10 sec.
RIPPLE is set as 1 except for the special case.
END becomes 'I' when A/T finishes. This Time P,I,D values are effective.
S_TIME should be same with PID7_CAL Event (In example 10 sec)
Sets the zooming level
GM7 Stop
R12,C4 Edit


Project Program Edit Toolbox Compile Online Debug Window Help
Be sure to use PID7 AT.END
DA_OUT is input to
Use voltage transformer.
S_TIME should be same
with PID7_CAL Event
(In example 10 sec)
Sets the zooning level
GM7 Stop
R18,C0
Edit

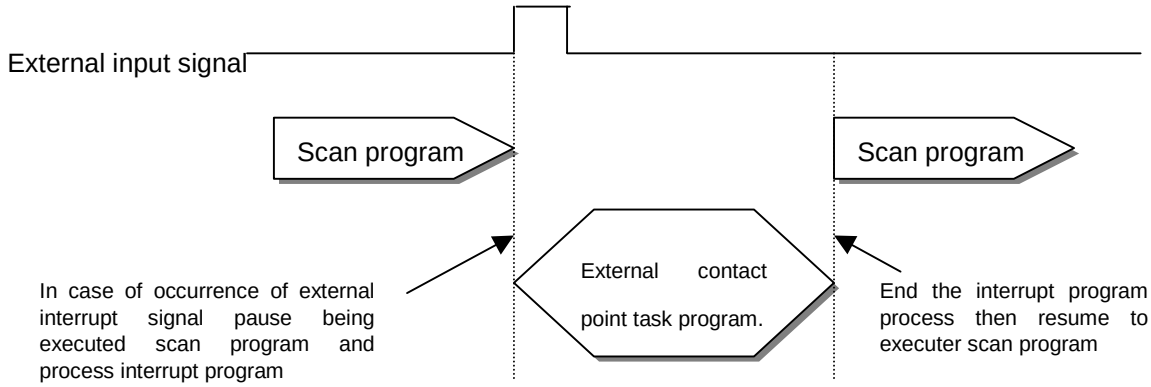
7.1.6 External Interrupt Function

In GM7 Series can perform max 8 points of external contact task by using input of base unit without special interrupt module.

1) Usage

This function is useful to execute a task program has been set to an external input signal.

2) Operating Explanation


External input signal
Scan program
Scan program
External contact point task program.
In case of occurrence of external interrupt signal pause being executed scan program and process interrupt program
End the interrupt program process then resume to executer scan program

3) Function

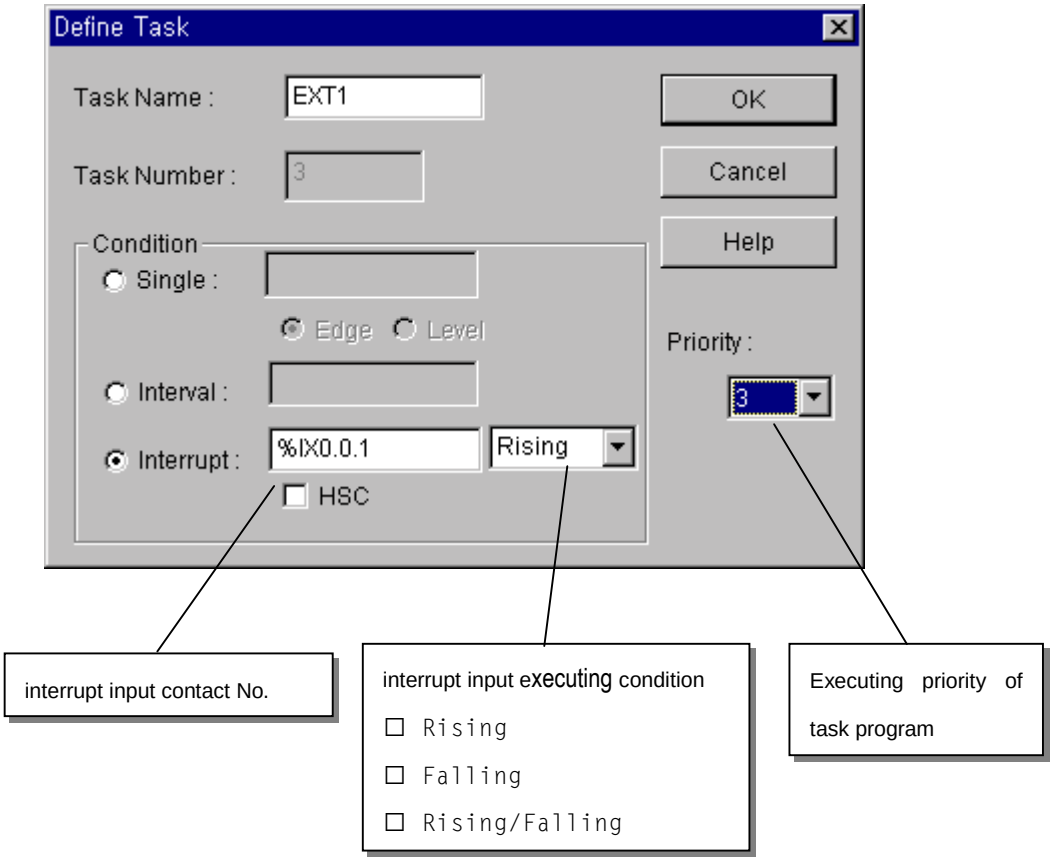
(1) Max. 8 points can be used to external interrupt input within %IXO.0.0 to%IX0.0.7.

(2) Inputting 8points of base unit are set functions like following.

	00	01	02	03	04	05	06	07
High speed counter	A-phase Input	B-phase Input	Preset Input	-	-	-	-	-
External interrupt task	☐	☐	☐	☐	☐	☐	☐	☐
Time driven task	-	-	-	-	-	-	-	-
Internal interrupt task	-	-	-	-	-	-	-	-
8points are available								

(3) Max, 8points of external contact point task are available to use. But the no. of them is decreased by using other task.

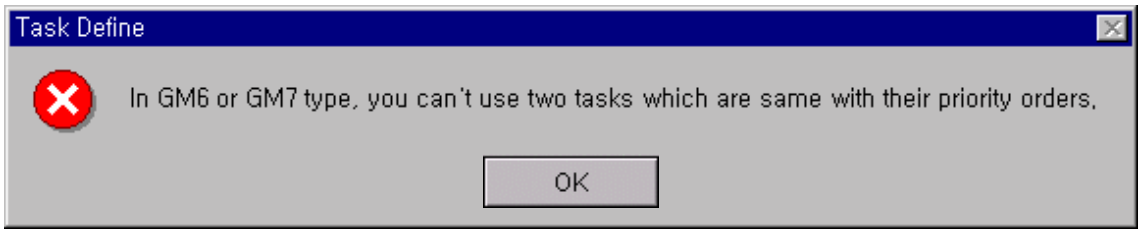
(4) Designate contact point, no. of priority and movement condition of the task program which is moved by interrupt inputting.



(5) Execute priority of task program. For the details, refer to GMWIN manual.

Remark

1) In case of GM7, executing priority of task can not be set equally, the following message occurs if the priority is set equally.



7. 2 Special module

7.2.1 A/D · D/A Combination module

1) Performance specification

The performance specification of the analog mixture module are following.

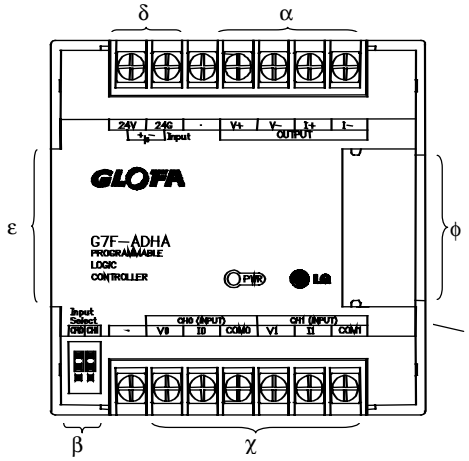
Item		Specifications		
Analog Input	Input range	Voltage	DC 0□10V (input resistance more than 1□)	
		Current	DC 0□20□ (input resistance 250Ω)	Classified by function block
			DC 4□20□ (input resistance 250Ω)	
	Digital output	12Bit(-48~4047)		
	Voltage/Current selection	1.Setting by jumper pin for V/I selection on upper part of product (Up: voltage, Down: Current) 2. Voltage/current selected by the program 3. When current input is used, short the V and I terminal		
	No. of channel	2Channels/1module		
	Absolute max. input	Voltage	DC +12V	
		Current	DC +24□	
Analog output	Output range	voltage	DC 0□10V (External load resistance 2□□1□)	
		Current	DC 0□20□ (External load resistance 510Ω)	Classified by function block
			DC 4□20□ (External load resistance 510Ω)	
	Digital Input	12Bit(-48~4047)		
	Voltage/Current selection	Separated from terminal		
	No, of channel	1Channel/1module		
	Absolute max. output	Voltage	DC +12V	
		Current	DC +24□	
Common	Max. resolution	Voltage	DC0□10V	2.5□ (1/4000)
		Current	DC0□20□	5□ (1/4000)
			DC4□20□	6.25□ (1/3200)
	Accuracy	±0.5% [Full scale]		
	Max. conversion speed	2□/CH + scan time		
	Insulation	Photo coupler insulation between I/O terminals and PLC power supply (No isolation between channels)		
	Connect terminal	9 Points 2 terminals		
	Internal current Consumption	20□		
	External power supply	DC 21.6 □ 26.4V, 80□		
	Weight	240g		

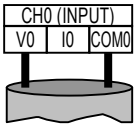
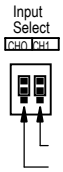
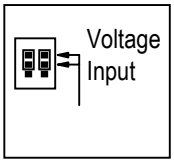
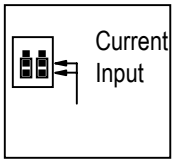
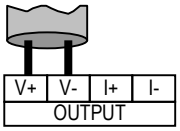
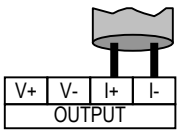
Remark

1) Offset/gain value can't be changed, it is fixed.
2) Analog inputting is set the current since this is manufactured.
3) Extend to use max.2 Modules

2) Names of parts and functions

Explain about names of parts and functions



No	Contents.
☐	RUN LED Indicate the operating status the G7F-ADHA
☐	Analog input terminal Voltage Input  Current input When current input is used, short the V and I terminal.
☐	Jumper pin of analog input  Right is CH.1selecting left is CH. 0 selecting  Connect upper parts by jumper pins  Connect lower parts by jumper pins.
☐	Analog output terminal Voltage output  Current output  ☐ Only one type of output (Current or Voltage)is available on a module
☐	External power input terminal ☐ Terminal supplies 24VDC
☐	Extension cable ☐ This cable is used to connect while analog mixture module is used..
☐	Extension cable connector The connector connects extension cable when extended module is used.

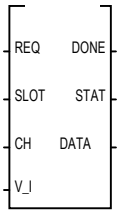
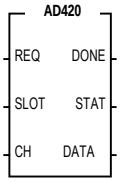
3) Function block

(1)Type of function block and funtion

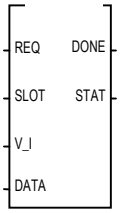
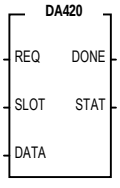
Function block	Function	Remark
ADHA_RD	Reading A/D conversion value	DC 0☐10V / DC 4☐20☐ Input only
AD420		DC 4☐20☐ current input only
DAHA_WR	Writing D/A conversion value	DC 0☐10V / DC 4☐20☐ output only
DA420		DC 4☐20☐ current output only

(2) Reading A/D conversion value (ADHA_RD, AD420)

Single type of function block for reading the module is performed for only one channel and the specified channel is used to read output variable of data displayed from A/D converted digital value.

Types of function block	Classification	Variable	Data type	Contents
	Input	REQ	BOOL	Execution request region of function block ● If connected condition on then region is completed and 0 turns to 1, then function block of reading module is executed while the program is performing
		SLOT	USINT	Location no. of slot ● Setting range:1 to 3
		CH	BOOL	Designation region of using channel ● Setting range:0to1
		V_I	BOOL	Designation region of Analog input type. Setting range:0 or 1(0: Current selecting, 1:Voltage selecting) AD420 isn't used in function block.
	Output	DONE	BOOL	Indicating region of A/D conversion value. ● If reading function block is completed to execute without an error then 1 is output and maintains 1 until next execution comes, but if an error occurs, 0 is output and if becomes operation stop status.
		STAT	USINT	Area marking error status ● When error occurs, output error numbers.
		DATA	INT	Area outputting A/D conversion value ● Data output range: -48 ☐ 4047

(3) Writing D/A conversion value (DAHA_WR, DA420)

Type of function block	I/O	Variables	Data type	Contents
	Input	REQ	BOOL	Execution request region of function block ● If connected condition on this region is completed and 0 turns to 1 then function block of writing module is executed while the program is performing.
		SLOT	USINT	Location no. of slot ● Setting range:1 to 3
		V_I	BOOL	Designation region of analog output type ● Setting range:0 or 1(0: I selecting, 1: V selecting) DA420 isn't used in function block.
		DATA	INT	Input region of D/A conversion ● Setting range:0 to 4000
	Output	DONE	BOOL	Indicating region of function block ● If writing function block is completed to execute without an error then 1 is output and maintains 1 until nest execution comes, but if an error occurs, 0 is output and it becomes operation stop status
		STAT	USINT	Area for marking error status, that outputs error number when error occurs in execution of function block.

4) Wiring

(1) Caution for wiring

☐Make sure that external input signal of the mixture module of AC and analog I/O is not affected by induction noise or occurs from the AC through using another cable.

☐Wire is adopted with consideration about peripheral temperature and electric current allowance. Thicker than Max. size of wire AWG22 (0.3☐) is better.

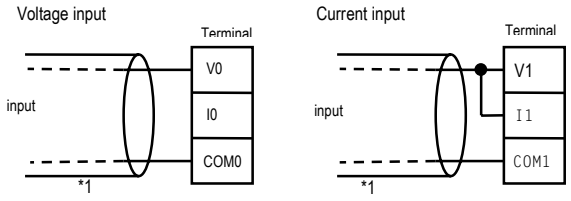
☐If wire is put near to high temp. radiated device or contacted with oil for a long time, it may cause of electric leakage so that it gets broken or miss-operation during wiring.

☐Be sure to connect with care of polarity while connecting to external 24V DC power supply.

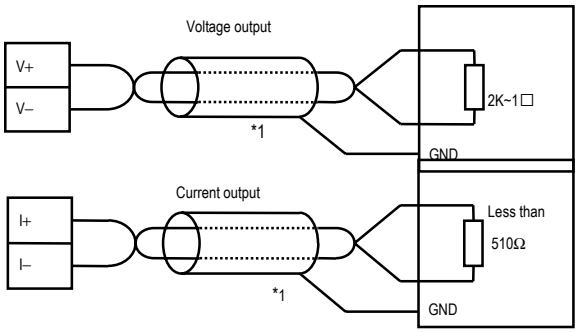
☐In case of wiring with high voltage line or generation line, it makes induction failure so then it may cause of miss-operation and out of order.

(2) Wiring example

☐) Analog input



☐) Analog output



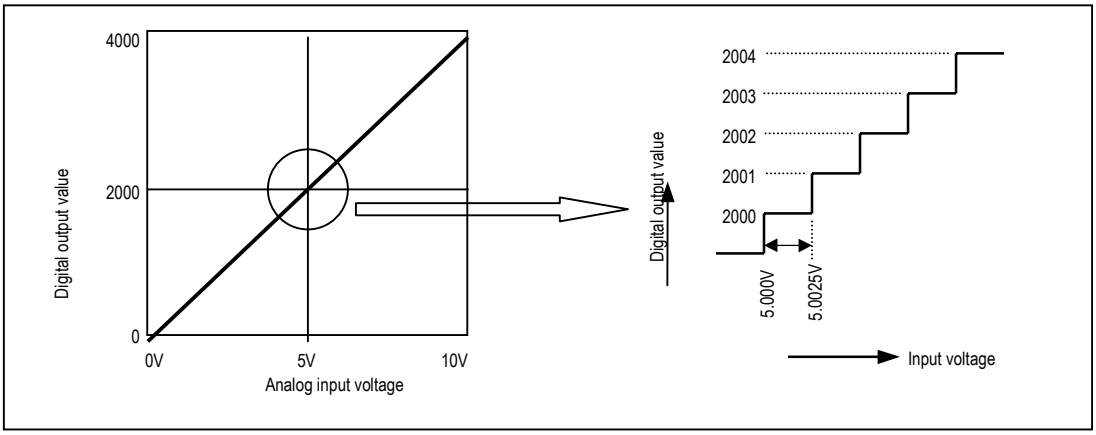
*1 : Be sure to use two-core twisted shield wire.

* Be careful to use that analog output is 1 channel.

5) I/O converstion characteristics

(1) Analog input characteristics

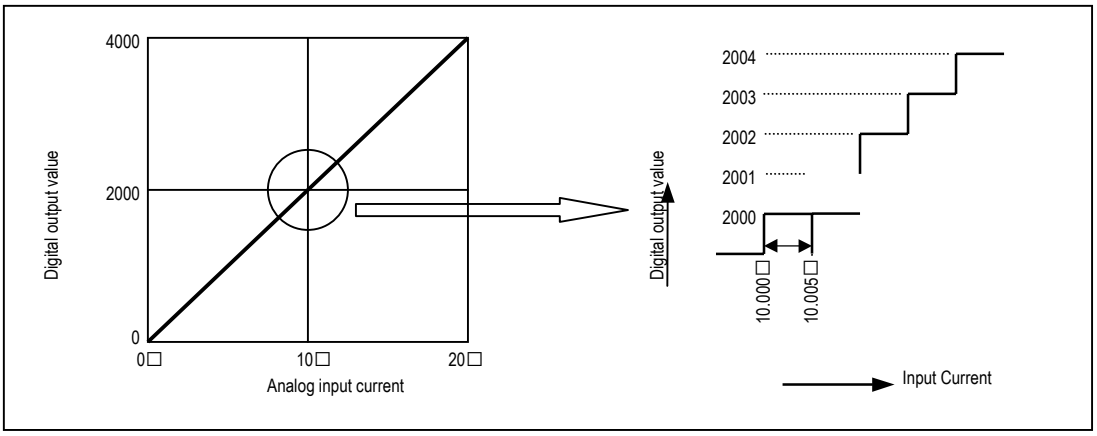
a) Voltage input



A/D conversion characteristics (voltage input)

In voltage input, digital amount 0 is output by 0V input and 4,000 is output by 10V input. Therefore input 2.5mV equals to digital amount 1, but value less than 2.5mV can't be converted.

b) Current input

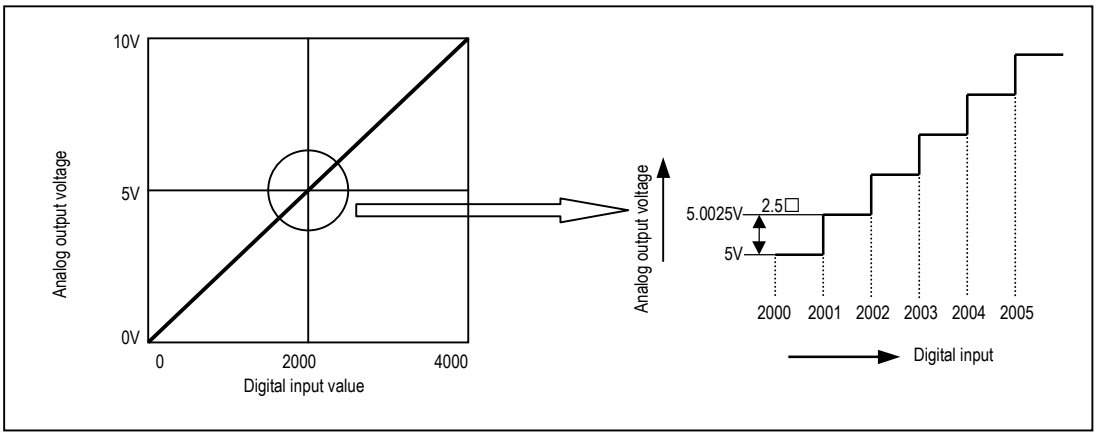


A/D conversion characteristics (Current input)

Current input 0mA becomes output 0, 10mA does 2000 and 20mA does 4000. therefore input 5☐ equals to digital amount 1, but value less tan 5☐ can't be converted. So abandon it.

(2) Analog output characteristics

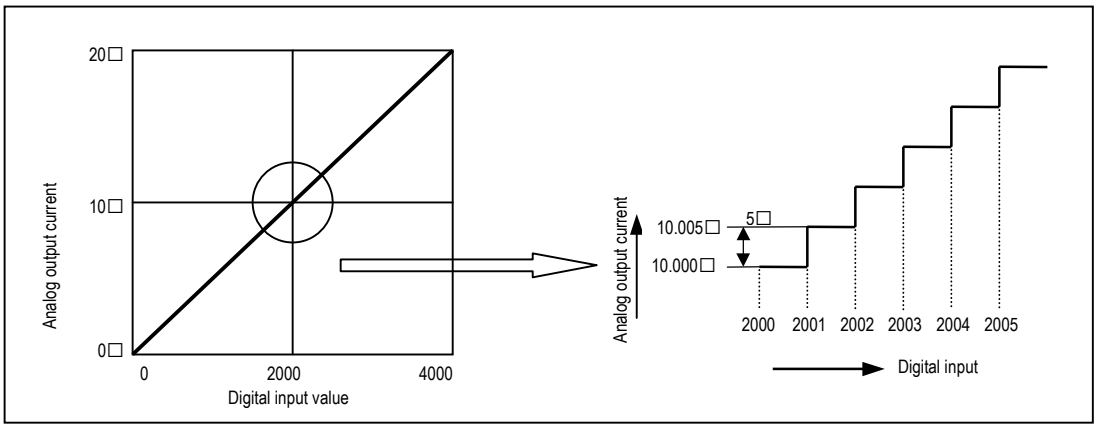
a) Voltage output



D/A conversion characteristic (voltage output)

Input of digital amount 0 outputs analog amount 0V, 4000 does 10V. Digital input 1 equals to 2.5mV of analog amount.

b) Current output



D/A conversion characteristic (Current output)

In current output, digital amount 0 exchanges to 0mA, and 4,000 does 20mA. Analog amount of digital input 1 equals to 5☐.

5) Program example

(1) Distinction program of A/D conversion value

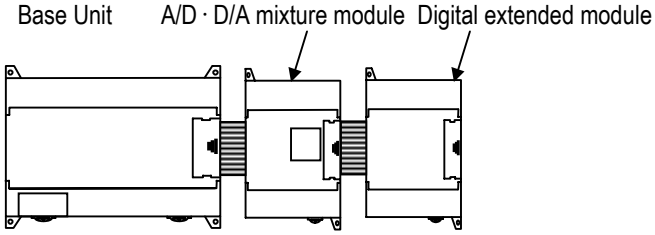
a) Program explanation

-When digital value of channel 0 is less than 2000, %Q0.2.0 is on.

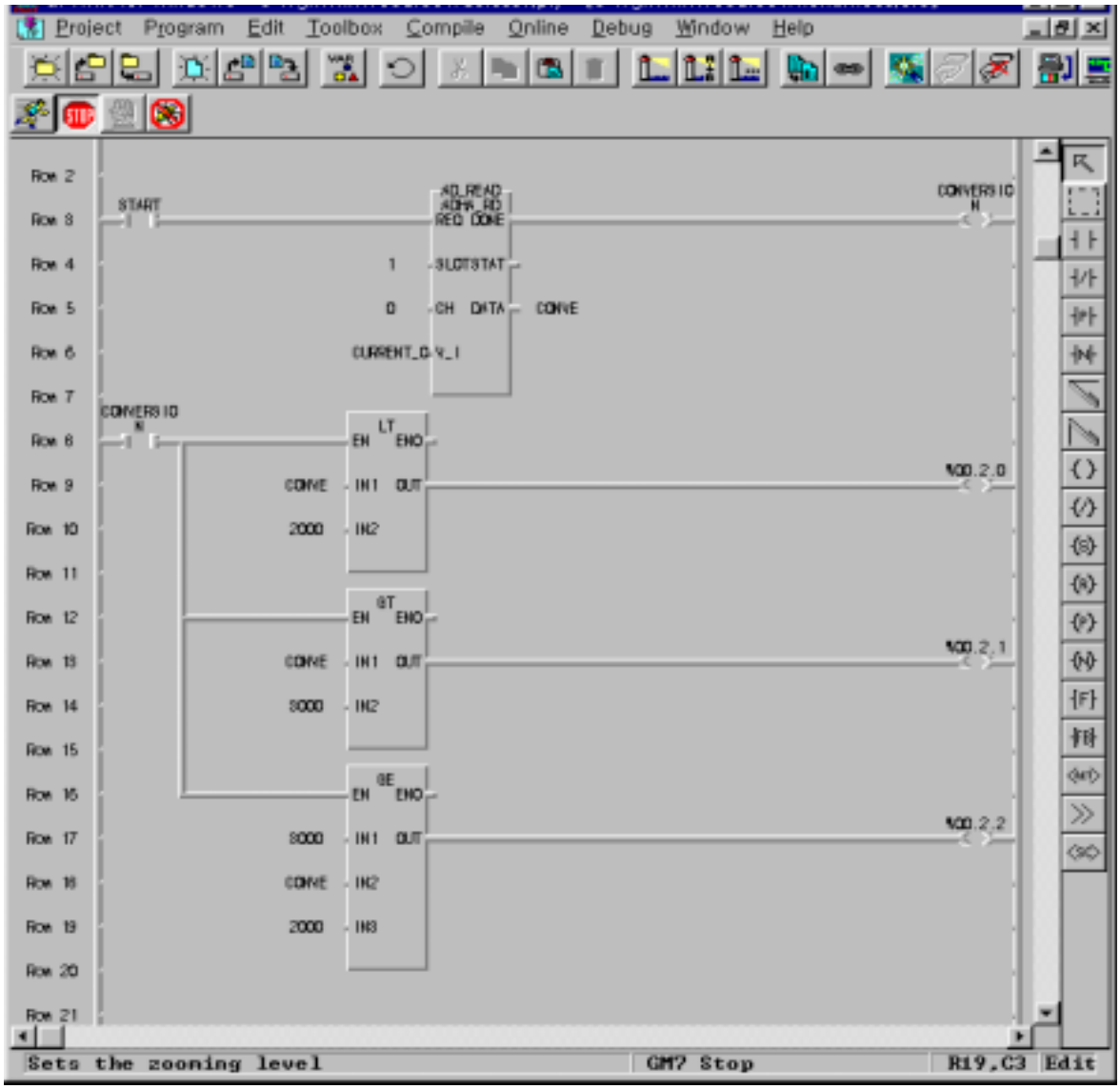
-when digital value of channel 0 is more than 3000, %Q0.2.1 is on.

-When digital value of channel 0 is more or same than 2000 or lesser than 3000, %Q0.2.2 is on.

b) System configuration



c) Program



(2) Program which controls speed of inverter by analog output voltage of 5 steps

a) Program explanation

-.When %IX0.0.0 becomes On, 2000 (5V) is output.

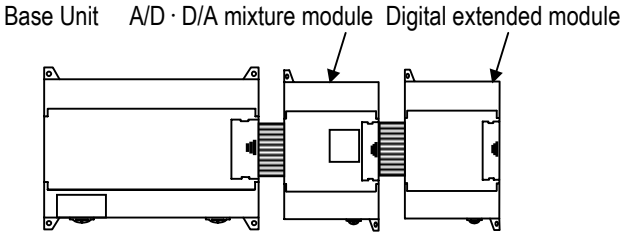
-. When %IX0.0.1 becomes On, 2400 (6V) is output.

-.When %IX0.0.2 becomes On, 2800 (7V) is output.

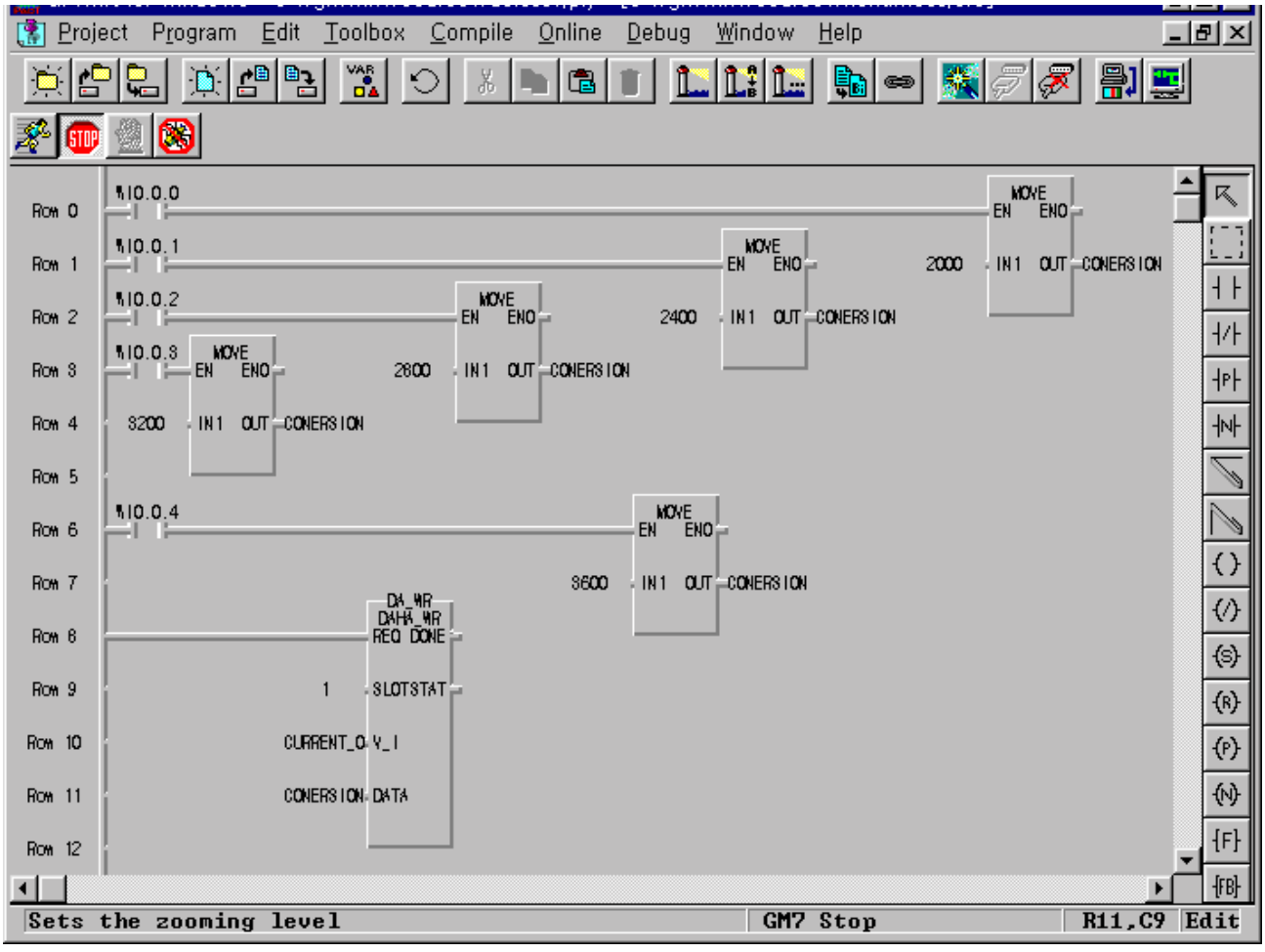
-.When %IX0.0.3 becomes On, 3200 (8V) is output.

-.When %IX0.0.4 becomes On, 3600 (9V) is output.

b) System configuration



c) Program



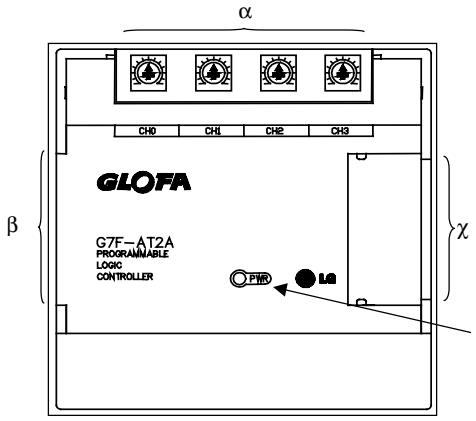
7.2.2 Analog timer

1) Performance specification

The performance specification of the analog timer module are following.

Item	Specification
Number of channels	4
Output value range	8 Bit (Digital output range: 0 ☐ 200)
Setting type	Setting by variable resistance
Accuracy of timer	±2.0% (Accuracy about max. value)
Operation method	Reading by means of using executive function block (AT2RD)
Internal current consumption	50☐
Number of module installment	Max 3 modules
Weight	200g

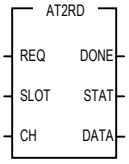
2) Names of parts and functions



No.	Name	Contents
☐	RUN LED	Indicate the operating status the G7F-AT2A. On: normal operating Off: DC 5V power off or the g7f-at2a module fault
☐	Channel	Setting up the length of timer through variable resistance to every channel.
☐	Extension cable	
☐	Extension cable connection terminal	

3) Function block

(1) Function of the function block

Function block type	Classification	Variable	Data type	Contents
	Input	REQ	BOOL	Executing request region of function block. ☐If condition, which is connected this region, is completed, then 0 becomes 1 and function block gets executed.
		SLOT	USINT	location no of slot ☐Region to write slot no. which analog timer module is installed. ☐Setting range: 1 to 3.
		CH	USINT	Designating region of using channel ☐Designating region for no. of using timer. ☐Setting range: 0 to 3
	Output	DONE	BOOL	Complete status of function block execution. ☐When input condition of function block is 1, if it is completed to execute, then '0☐1' is output. But when input condition is 0 and output changes to '1☐0,' then related timer becomes operating stop status.
		STAT	USINT	Indication region of error status ☐Region for output of error no. when error occurs while function block execution.
		DATA	USINT	Digital value output of timer. Output range: 0 to 200.

(2) Error list, which is occurred on, output variable STAT

No.	Contents	Remedy
3	Setting range excess of slot location no.	Designate right slot no. which is installed with analog timer module.
4	Empty analog timer module of designated slot.	Installing analog timer module to designated slot.
5	Another module is installed which is different from analog timer module.	Installing analog timer module to designated slot.
6	Setting region excess of channel	Correct using channel no. designation.

4) Variable resistance handling

(1) Set up extension G7F-AT2A to CPU module and allow power supply then make program by means of analog timer module executive function block AT2RD, after that download it to CPU module.

(2) Change CPU module to RUN mode and monitor output variable DATA value of function block AT2RD then control variable resistance of channel which is set up on function block of G7F-AT2A.

(3) Then output value becomes small if variable resistance turns left. And value of DATA becomes big, if it turns right.

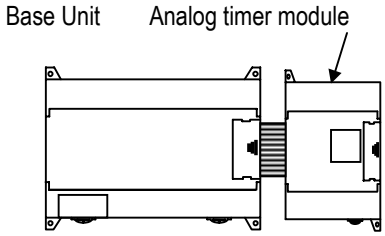
(4) When expected timer value is output as data, stops controlling variable resistance.

5) Program example

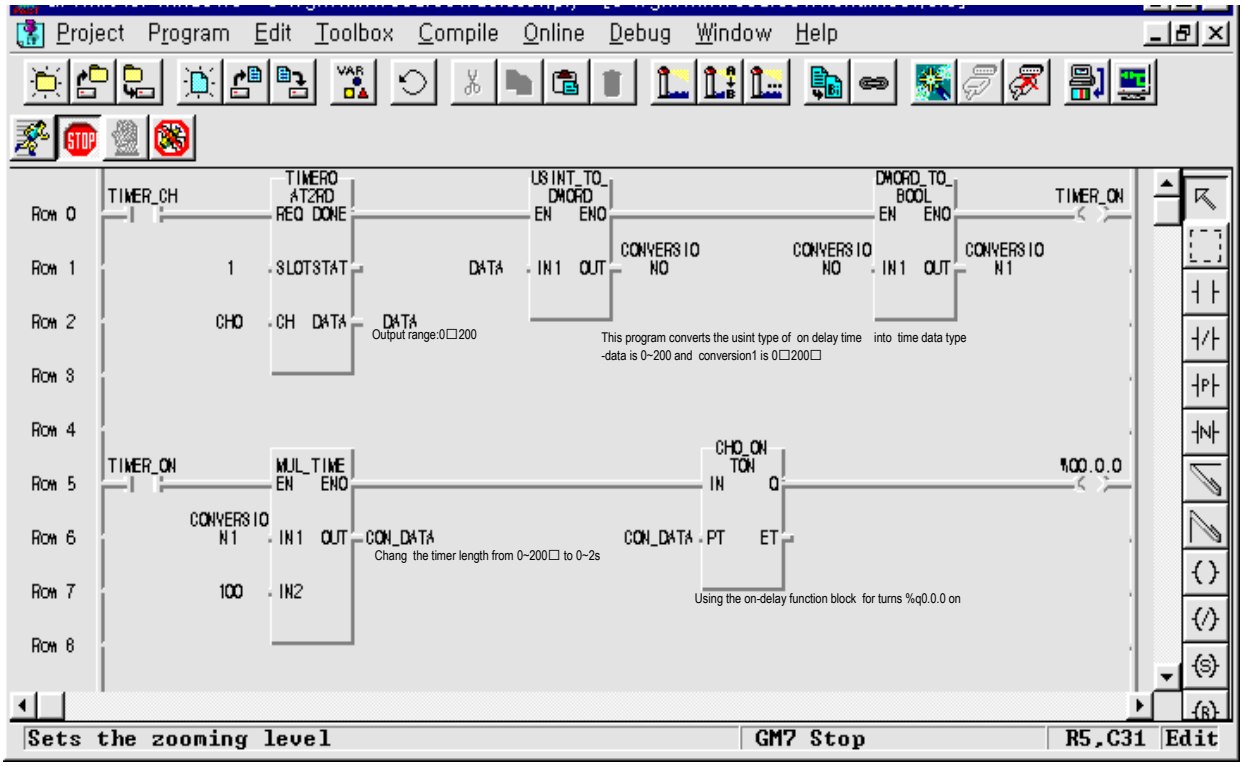
(1) Program explanation

Program which controls on-delay time of output contact point within 0 to 20 sec. By analog timer module.

(2) System configuration



(3) Program



Chapter 8 Communication Function

8.1 Dedicated Protocol Communication

8.1.1 Introduction

GM7's built-in Cnet communication uses only GM7 base unit for a dedicated communication. That is, it doesn't need a separate Cnet I/F module to facilitate the user-intended communication system by utilizing reading or writing of any area in CPU, and monitoring function.

GM7 base unit serves as follows:

- Individual/continuous reading of direct variable
- Individual/continuous writing of direct variable
- Reading CPU status
- Monitor variables registration
- Executing monitoring
- 1:1 connection(link between G SERIES's) system configuration (GM7 base unit: RS-232C)

Remark

GM7 built-in communication function supports Cnet communication without any separate Cnet module. It must be used under the following instructions.

1) GM7 base unit supports 1:1 communication only. for 1:N system having master-slave Format, use GM7 base unit with G7L-CUEC module connected. G7L-CUEC module supports RS-422/485 protocol.
2) RS-232C communication cable for GM7 base unit is different from RS-232C cable for GMWIN in pin arrangement and from the cable for Cnet module, too. The cable can't be used without any treatment. For the detailed wiring method, refer to 8.1.2.
3) It's possible to set baud rate type and M area size in GMWIN. For the detailed information, refer to the appendix or GMWIN manual.

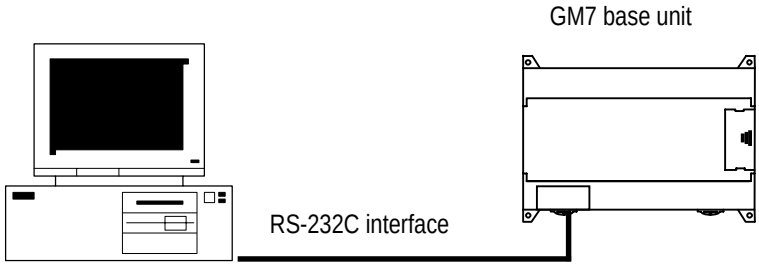
8.1.2 System configuration method

According to the method of connection, the system using GM7 built-in communication can be composed.

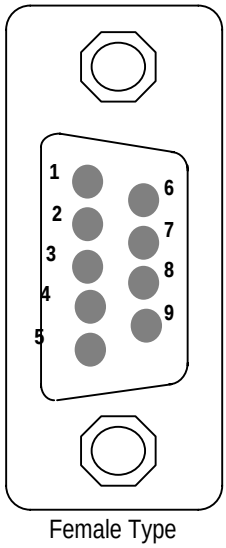
1) Connecting system configuration (link between G SERIES's)

(1) 1:1 connection with general PC

① Communication program made by C or BASE computer language on the user's computer, or utility program like MMI software can be used.



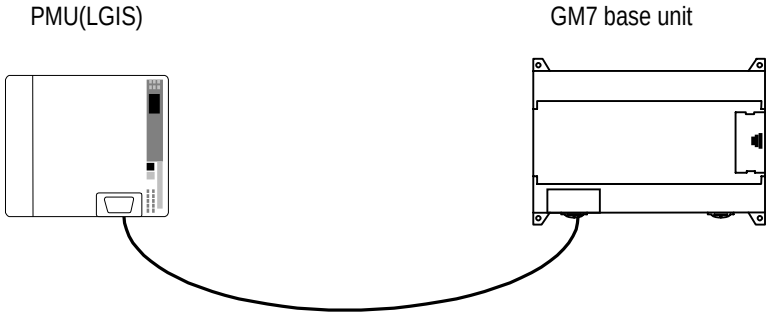
② Wiring method

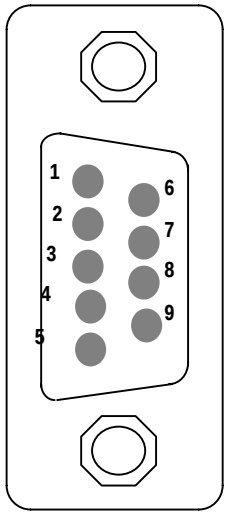


Female Type

PC	Pin assignment And direction	GM7 base unit	
Pin No.		Pin No.	Signal
1		1	CD
2		2	RXD
3		3	TXD
4		4	DTR
5		5	SG
6		6	DSR
7		7	RTS
8		8	CTS
9		9	RI

(2) 1:1 connection with a monitoring device like PMU





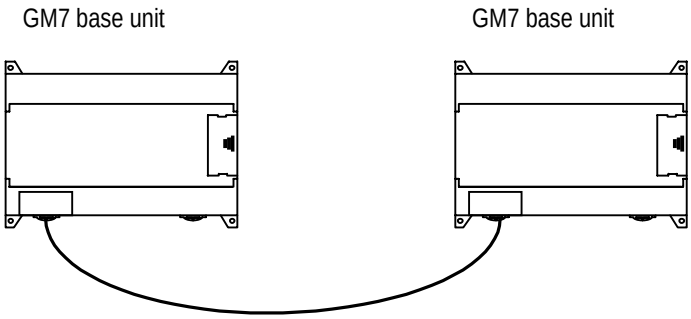
Female Type

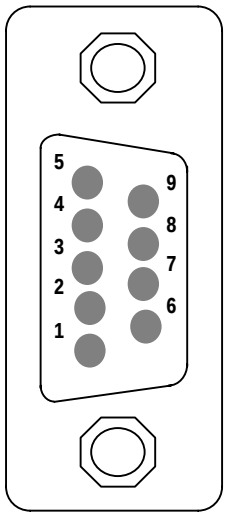
PMU	Pin assignment and direction	GM7 base unit	
Pin No.		Pin no.	Signal
1		1	CD
2		2	RXD
3		3	TXD
4		4	DTR
5		5	SG
6		6	DSR
7		7	RTS
8		8	CTS
9		9	RI

.

(3) 1:1 connection with other GM7

For the detailed inFormation, refer to 8.1.7 “1:1 Dedicated Protocol Communication.”





Male Type

GM7 base unit	Pin assignment and direction	GM7 base unit	
Pin no.		Pin no.	Signal
1		1	CD
2		2	RXD
3		3	TXD
4		4	DTR
5		5	SG
6		6	DSR
7		7	RTS
8		8	CTS
9		9	RI

8.1.3 Frame Structure

1) Base Format

(1) Request frame(external communication device GM7 base unit)

(Max. 256 Bytes)

Header (ENQ)	Station	Command	Command type	Structurized data area	Tail (EOT)	Frame check (BCC)

(2) ACK Response frame (GM7 base unit external communication device, when receiving data normally)

(max. 256 Bytes)

Header (ACK)	Station	Command	Command type	Structurized data area or null code	Tail (ETX)	Frame check (BCC)

(3) NAK Response frame (GM7 base unit external communication device, when receiving data abnormally)

(max. 256 Bytes)

Header (NAK)	Station	Command	Command type	Error code (ASCII 4 Byte)	Tail (ETX)	Frame check (BCC)

Remark

1) Used control codes are as follows. Be familiar with the following control codes. Because they are importantly used for communication.

[Control codes]

Codes	Hex value	Name	Contents
ENQ	H05	Enquire	Request frame initial code
ACK	H06	Acknowledge	ACK response frame initial code
NAK	H15	Not Acknowledge	NAK response frame initial code
EOT	H04	End of Text	Request frame ending ASCII code
ETX	H03	End Text	Response frame ending ASCII code

Remark

1) The numerical data of all frames are ASCII codes equal to hexadecimal value, if there's no clear statement. The terms in hexadecimal are as follows.

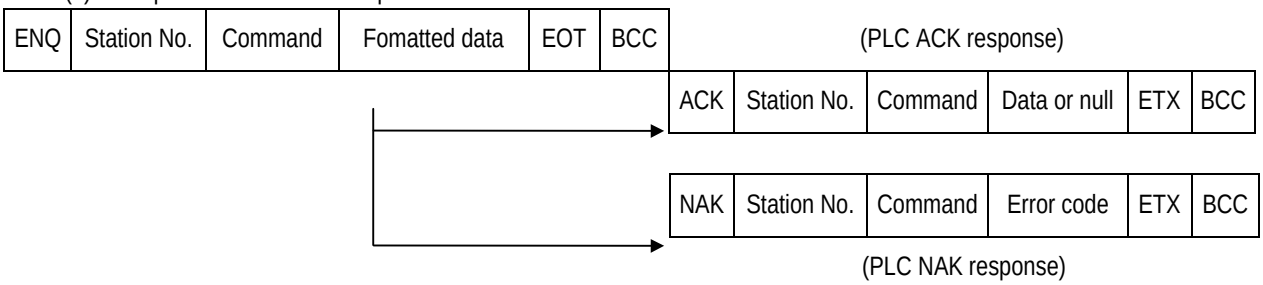
- Station No.
- When the main command is R(r) or W (w) and the command type is numerical (means a data type)
- All of the terms indicating size of all data in the Formatted data area.
- Monitoring registration and command registration number of execution commands.
- All contents of data

Remark

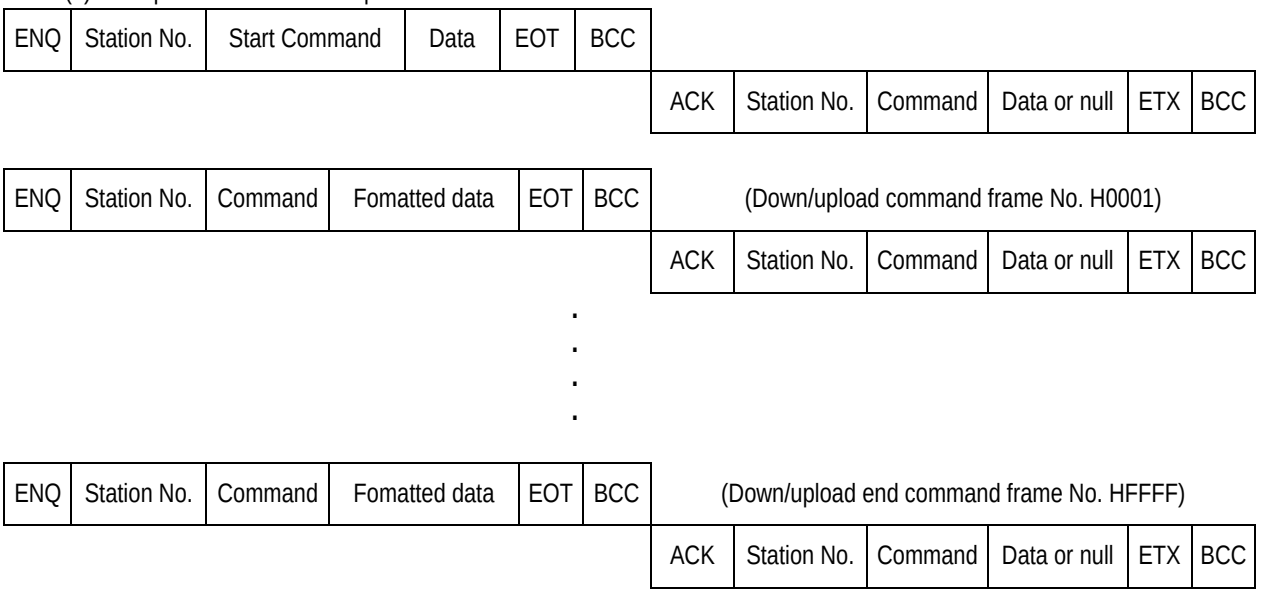
1) If it is hexadecimal, H is attached in front of the number of frames like H01, H12345, H34, H12, and H89AB.

1) Sequence of command frame

(1) Sequence of command request frame



(2) Sequence of Download/upload frame



.
.
.
.

8.1.4 List of commands

Command list for direct communication.

Division / Item		Command: Main command Code	Command: Main command ASCII code	Command: Command type Code	Command: Command type ASCII code	Treatment
Reading direct variable	Indivisual reading	r(R)	H72 (H52)	SS	5353	Reads direct variables of Bit, Byte, Word and Dword type.
	Continuos reading	r(R)	H72 (H52)	SB	5342	Reads direct variables of Byte, Word and Dword type in block unit.(Continuous reading Bit is unavailable)
Writing direct variable	Indivisual writing	w (W)	H77 (H57)	SS	5353	Writes data to direct variable of Bit, Byte, Word and Dword type.
	Continuos writing	w(W)	H77 (H57)	SB	5342	Writes data to direct variable of Byte, Word and Dword type in block unit. (Continuous reading Bit is unavailable)
CPU Status reading		r(R)	H73 (H53)	ST	5354	Reads flag list like PLC operation status and error inFormation. (For detailed flag contents, refer to GM7 manual).

Division / Item	Command: Main command Code	Command: Main command ASCII code	Command: Register No.	Command: Register No. ASCII code	Treatment
Monitoring variable register	x(X)	H78 H58	H00~H09	3030 ~ 3039	Register variables to monitor.
Execution of monitoring	y(Y)	H79 (H59)	H00~H09	3030 ~ 3039	Execute registered variables to monitor.

Remark

1) GM7 base unit identifies capitals or small letters for main commands, but not for the others.

2) If it's a main command in capitals, it calculates BCC value. But if it's not, it doesn't. Therefore, when BCC for frame check is used, main commands must be in small letters.

3) Commands to read/write symbolic variable or array variable can't be used in built-in communication

8.1.5 Data type

It's possible to read and write direct variable in built-in communication. When direct variable is used, be aware of data type.

1) Data type of direct variable

- Available types of device: M (internal memory), Q (output), I (input)
- When direct variable is used, attach '%'(25H) in front of the marking characters.

Data type	Marking characters	Examples
Bit	X(58H)	%MX0, %QX0.0.0, %IX0.0.0
Byte	B(42H)	%MB10, %QB0.0.0, %IB0.0.0
Word	W(57H)	%MW10, %QW0.0.0, %IW0.0.0
Double Word	D(44H)	%MD10, %QD0.0.0, %ID0.0.0

Remark

1) Commands to read/write symbolic variable can't be used in built-in communication.

2) Memory address 100 in '%MB100' indicates decimal value.

3) Long word among data types can't be used.

8.1.6 Execution of commands(Ex.)

1) Individual reading of direct variable(RSS)

(1) Introduction

This is a function that reads PLC device memory directly specified in accord with memory data type. Separate device memory can be read up to 16 at a time.

(2) PC request format

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable length	Variable name	.....	Tail	Frame check
Ex. of frame	ENQ	H20	R(r)	SS	H01	H06	%MW100		EOT	BCC
ASCII value	H05	H3230	H52(72)	H5353	H3031	H3036	H254D57313030		H04	

1 block (setting can be repeated up to max. 16 block(

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII and added to BCC. For example, the BCC of the above frame is gotten as below: H05+H32+H30+H72+H53+H53+H30+H31+H30+H36+H25+H4D+H57+H31+H30+H30+H04 =H03A4 Therefore BCC is A4.
Number of Blocks	This specifies how much of the blocks composed of "[Variable length][Variable name]" are in this request Format. This can be set up to 16. Therefore, the value of [Number of blocks] must be set between H01(ASCII value:3031)-H10(ASCII value:3030).
Variable length(Name length of direct variable)	This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130). For example, if the variable name is %MWO, it has 4 characters to be H04 as its length. If %QW0.0.0, 8 characters to be H08.
Variable name	Address to be actually read is entered. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered.

Remark

1) Numerical data of frame(Ex.) is hex value, and "H" is unnecessary during preparing real frame.
2) Device data type of each must be same. If data type of the first block is WORD, and the second block is DOUBLE WORD, error occurs.

(3) Response format (ACK response)

Format name	Header	Station No.	Command	Command type	Number of blocks	Number of data	data	.....	Tail	Frame check
Ex. of frame	ACK	H20	R(r)	SS	H01	H02	HA9F3		ETX	BCC
ASCII value	H06	H3230	H52(72)	H5353	H3031	H3032	H41394633		H04	

1 block(max. 16 blocks)

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII and added to BCC, and sent.
Number of data	Number of data means byte number of hex type, and is converted into ASCII. This number is determined according to memory type(X,B,W,D) included in direct variable name of computer request Format. Number of data in accordance with its variable type is as follows:
Data	• In data area, there are the values of hex data converted to ASCII code saved.

Data type	Available direct variable	Number of data
BOOL(X)	%MX,%QX,%IX	1
Byte(B)	%MB,%QB,%IB	1
WORD(W)	%MW,%QW,%IW	2
DOUBLE WORD(D)	%MD,%QD,%ID	4

Ex.1

The fact that number of data is H04(ASCII code value:H3034) means that there is hex data of 4 bytes in data(DOUBLE WORD). Hex data of 4 bytes is converted into ASCII code in data.

Ex.2

If number of data is H04 and the data is H12345678, ASCII code converted value of this is "31 32 33 34 35 36 37 38," and this contents is entered in data area. Namely, highest value is entered first, lowest value last.

Remark

1) If data type is BOOL, data read is indicated by bytes of hex. Namely, if Bit value is 0, it indicated by H00, and if 1, by H01.

(4) Response format (NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Byte)	Tail	Frame check
Ex. of frame	NAK	H20	R(r)	SS	H1132	ETX	BCC
ASCII value	H15	H3230	H52(72)	H5353	H31313332	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC.
Error code	Hex and 2 bytes(ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes when 1 WORD from %MW20 and 1 WORD from %QW0.2.1 address of station No.1 are read and BCC value is checked. Also it is supposed that H1234 is entered in %MW20, and data of H5678 is entered in %QW0.2.1.

① Computer request format (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable length	Format name	Variable length	Format name	Tail	Frame check
Ex. of frame	ENQ	H01	r	SS	H02	H05	%MW20	H08	%QW0.0.1	EOT	BCC
ASCII value	H05	H3031	H72	H5353	H3032	H3035	H254D573230	H3038	H255157302E322E31	H04	

② For ACK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Number of blocks	Number of data	Data	Number of data	Data	Tail	Frame check
Ex. of frame	ACK	H01	r	SS	H02	H02	H1234	H02	H5678	ETX	BCC
ASCII value	H06	H3031	H72	H5353	H3032	H3032	H31323334	H3032	H35363738	H03	

③ For NAK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Error code	Tail	Frame check
Ex. of frame	NAK	H01	r	SS	Error code (2)	ETX	BCC
ASCII value	H15	H3031	H72	H5353	Error code (4)	H03	

Frame check BCC is automatically calculated internally.

2) Continuous reading(RSB) of direct variable

(1) Introduction

This is a function that reads the PLC device memory directly specified in accord with memory data type. With this, data is read from specified address as much as specified continuously.

(2) PC request format

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable length	Number of data (Max. 128 Bytes)	Tail	Frame check
Ex. of frame	ENQ	H10	R(r)	SB	H06	%MW100	H05	EOT	BCC
ASCII value	H05	H3130	H52(72)	H5342	H3036	H255744313030	H3035	H04	

Remark

1) Number of data specifies the number according to the type of direct variable. Namely, if the data type of direct variable is double word, and number is 5, it means that 5 DOUBLE WORDs should be read.
2) Max. of %MB in number of data can be used up to 128 . %MW up to 64 and %MD, 32.
3) Protocol of RSB doesn't have number of blocks.

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII and added to BCC.
Variable length(Name length of direct variable)	This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).
Variable name	Address to be actually read is entered. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered.

(3) GM7 Base Unit response format (GM7 of ACK response)

Format name	Header	Station No.	Command	Command type	Number of blocks	Number of data	data	Tail	Frame check
Ex. of frame	ACK	H10	R(r)	SB	H01	H02	H1122	EOT	BCC
ASCII value	H06	H3130	H52(72)	H5342	H3031	H3134	H31313232	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII and added to BCC, and sent.
Number of data	means byte number of hex type, and is converted into ASCII. This number is determined by multiplying the data number of computer request Format by the data size(in below table) according to memory type(B,W,D) included in direct variable name of computer request Format.
Data	.In data area, there are the values of hex data converted to ASCII code saved.

Data type	Available direct variable	Data size
Byte(B)	%MB,%QB,%IB	1
WORD(W)	%MW,%QW,%IW,%(P,M,L,K,F,T,C,D,S)W	2
DOUBLE WORD(D)	%MD,%QD,%ID	4

Ex.1

When memory type included in direct variable name of computer request Format is W(WORD), and data number of computer request Format is 03, data number of PLC ACK response after execution of command is indicated by H06(2*03 = 06 bytes)Byte and ASCII code value 3036 is entered in data area.

Ex.2

In just above example, when data contents of 3 WORDs are 1234, 5678, and 9ABC in order, actual ASCII code converted values are 31323334 35363738 39414243, and the contents are entered in data area.

(4) Response format(NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Byte)	Tail	Frame check
Ex. of frame	NAK	H10	r	SB	H1132	ETX	BCC
ASCII value	H15	H3130	H72	H5342	H31313332	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC, and sent.
Error code	Hex and 2 bytes(ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes that 2 DOUBLE WORDs from %MDO of station NO. 10 is read and BCC value is checked. Also it is supposed that data in %MDO and in %MD1 is as follow:

%MDO = H12345678

%MD1 = H9ABCDEF0

¯Computer request format (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Data length	Variable name	Number of data	Tail	Frame check
Frame (Example)	ENQ	H0A	r	SB	H04	%MD0	H02	EOT	BCC
ASCII value	H05	H3041	H72	H5342	H3034	H254D4430	H3032	H04	

α For ACK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Number of data	data	Tail	Frame check
Frame (Example)	ACK	H0A	r	SB	H08	12345678 9ABCDEF0	ETX	BCC
ASCII value	H06	H3041	H72	H5342	H3038	H313233343536373839414 24344454630	03	

β For NAK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No>	Command	Command type	Error code	Tail	BCC
Frame (Example)	NAK	H0A	r	SB	Error code (2)	ETX	BCC
ASCII value	H15	H3041	H72	H5342	Error code (4)	H03	

3) Individual writing of direct variable(W(w)SS)

(1) Introduction

This is a function that writes the PLC device memory directly specified in accord with memory data type.

(2) PC request format

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable Length	Variable Name	Data	.....	Tail	Frame check
Frame (Example)	ENQ	H20	W(w)	SS	H01	H06	%MW100	H00E2		EOT	BCC
ASCII value	H05	H3230	H57(77)	H5353	H3031	H3036	H254D5731 3030	H30304 532		H04	

1 block(setting can be repeated up to max. 16 blocks)

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII and added to BCC.
Number of blocks	This specifies how much of the blocks composed of "[Variable length][Variable name]" are in this request Format. This can be set up to 16. Therefore, the value of [Number of blocks] must be set between H01(ASCII value:3031)-H10(ASCII value:3030).
Variable length(Name length of direct variable)	This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).
Direct variable	Address to be actually read is entered. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered.
Data	If the value to be written in %MW100 area is H A, the data Format must be H000A. If the value to be written in %MD100 area is H A, the data Format must be H0000000A. In data area, the ASCII value converted from hex data is entered.

Ex.1

If type of data to be currently written is DOUBLE WORD, the data is H12345678, ASCII code converted value of this is "3132333435363738," and this content must be entered in data area. Namely, most significant value must be sent first, least significant value last.

Remark

1) Device data types of each block must be the same.

2) If data type is BOOL, the data to be written is indicated by bytes of hex. Namely, if Bit value is 0, it must be indicated by H00(3030), and if 1, by H01(3031).

(3) Response format (ACK response)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Example)	ACK	H20	W(w)	SS	ETX	BCC
ASCII value	H06	H3230	H57(77)	H5353	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII and added to BCC, and sent.

(4) Response format (NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Byte)	Tail	Frame check
Frame (Example)	NAK	H20	W(w)	SS	H4252	ETX	BCC
ASCII value	H15	H3230	H57(77)	H5353	H34323532	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC, and sent.
Error code	Hex and 2 bytes(ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes that "HFF" is written in %MW230 of station No. 1 and BCC value is checked.

¯Computer request format (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable Length	Variable Name	Data	Tail	Frame check
Frame (Example)	ENQ	H01	w	SS	H01	H06	%MW230	H00FF	EOT	BCC
ASCII value	H05	H3031	H77	H5353	H3031	H3036	H254D5732 3330	H30304646	H04	

α For ACK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Example)	ACK	H01	w	SS	ETX	BCC
ASCII value	H06	H3031	H77	H5353	H03	

β For NAK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Error code	Tail	Frame check
Frame (Example)	NAK	H01	w	SS	Error code (2)	ETX	BCC
ASCII value	H15	H3031	H77	H5353	Error code (4)	H03	

4) Continuous writing of direct variable(WSB)

(1) Introduction

This is a function that directly specifies PLC device memory and continuously writes data from specified address as much as specified length.

(2) Request format

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable Length	Number of data (Max.128 Byte)	Data	Tail	Frame check
Frame (Example)	ENQ	H10	W(w)	SB	H06	%MD100	H01	H11112222	EOT	BCC
ASCII value	H05	H3130	H57(77)	H5342	H3036	H254D44313030	H3031	H3131313132323232	H04	

Remark

1) Number of data specifies the number according to the type of direct variable. Namely, if the data type of direct variable is DOUBLE WORD, and number of data is 5, it means that 5 DOUBLE WORDs should be written.

2) Max. of %MB in number of data can be used up to 128, %MW, 64, and %MD, 32.

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII and added to BCC.
Variable length(Name length of direct variable)	This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).
Direct variable	Address to be actually read. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered.

Remark

1) Protocol of WSB doesn't have the number of blocks.

(3) Response Format (ACK response)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Example)	ACK	H10	W(w)	SB	ETX	BCC
ASCII value	H06	H3130	H57(77)	H5342	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII and added to BCC, and sent.

(4) Response Format(NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Byte)	Tail	Frame check
Frame (Example)	ENQ	H10	W(w)	SB	H1132	EOT	BCC
ASCII value	H05	H3130	H57(77)	H5342	H31313332	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC, and sent.
Error code	Hex and 2 bytes(ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes that 4 byte HAA15056F is written in %QD0.0.0 of station NO. 1 and BCC value is checked.

¯Computer request Format(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Variable Length	Variable Name	Number of data	Data	Tail	Frame check
Frame (Example)	ENQ	H01	w	SB	H08	%QD0.0.0	H01	HAA15056F	EOT	BCC
ASCII value	H05	H3031	H77	H5342	H3038	H254442302E302E30	H3031	H41413135603536	H04	

α For ACK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Example)	ACK	H01	W	SB	ETX	BCC
ASCII value	H06	H3031	H77	H5342	H03	

β For NAK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Error code	Tail	Frame check
Frame (Example)	NAK	01	W	SB	Error code (2)	ETX	BCC
ASCII value	H15	H3031	H77	H5342	Error code (4)	H03	

5) Monitor register (X##)

(1) Introduction

Monitor register can separately register up to 10 in combination with actual variable reading command, and carries out the registered one through monitor command after registration.

(2) PC request Format

Format name	Header	Station No.	Command	Registration No.	Registration Format	Tail	Frame check
Frame (Example)	ENQ	H10	X(x)	H09	Refer to registration Format	EOT	BCC
ASCII value	H05	H3130	H58(78)	H3039	[ ]	H04	

Item	Explanation
BCC	When command is one of lower case(x), only one lower byte of the value resulted by adding 1 byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC.
Register No.	This can be registered up to 10(0 to 9, H00-H09), and if an already registered No. is registered again, the one currently being executed is registered.
Register Format	This is used to before EOT in command of Formats of separate reading of direct variable, continuous reading, and named variable reading.

; Register Format: Register Format of request Formats must select and use only one of the followings.

① Individual reading of direct variable

RSS	Number of blocks(2 Byte)	Variable length (2 Byte)	Variable name (16 Byte)	...

1 block(max. 16 blocks)

② Continuous reading of direct variable

RSB	Variable length (2 Byte)	Variable name (16 Byte)	Number of data

(3) Response Format(ACK response)

Format name	Header	Station No.	Command	Registration No.	Tail	Frame check
Frame (Example)	ACK	H10	X(x)	H09	ETX	BCC
ASCII value	H06	H3130	H58(78)	H3039	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII and added to BCC, and sent.

(4) Response Format (NAK response)

Format name	Header	Station No.	Command	Registration No.	Error code (Hex 2Byte)	Tail	Frame check
Frame (Example)	ACK	H10	X(x)	H09	H1132	ETX	BCC
ASCII value	H06	H3130	H58(78)	H3039	H31313332	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC, and sent.
Error code	Hex and 2 bytes(ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes that direct variable %MWO of station NO. 1 is monitor registered.

¯Computer request Format (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Registration No.	Registration Format				Tail	Frame check
					R##	Number of blocks	Variable length	Variable name		
Frame (Example)	ENQ	H01	x	H01	RSS	H01	H04	%MW0	EOT	BCC
ASCII value	H05	H3031	H78	H3031	H525353	H3031	H3034	H41534446	H04	

α For ACK response after execution of command(PC GM7 Base Unit)

Format name	Header	Station No.	Command	Registration No.	Tail	Frame check
Frame (Example)	ACK	H01	x	H01	ETX	BCC
ASCII value	H06	H3031	H78	H3031	H03	

β For NAK response after execution of command (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Registration No.	Error code	Tail	Frame check
Frame (Example)	NAK	H01	x	H01	Error code (2)	ETX	BCC
ASCII value	H15	H3031	H78	H3031	Error code (4)	H03	

6) Monitor execution(Y##)

(1) Introduction

This is a function that carries out the writing of the variable registered by monitor register. This also specifies a registered number and carries out writing of the variable registered by the number.

(2) PC request Format

Format name	Header	Station No.	Command	Registration No.	Tail	Frame check
Frame (Example)	ENQ	H10	Y(y)	H09	EOT	BCC
ASCII value	H05	H3130	H59(79)	H3039	H03	

Item	Explanation
Register No.	Register No. uses the same number registered during monitor register for monitor execution. It is possible to set from 00-09(H00-H09).
BCC	When command is one of lower case(y), only one lower byte of the value resulted by adding 1 byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC.

(3) Response Format(ACK response)

① In case that the register Format of register No. is the Individual reading of direct

Format name	Header	Station No.	Command	Registration No.	Number of Blocks	Number of data	Data	Tail	Frame check
Frame (Example)	ACK	H10	Y(y)	H09	H01	H04	H9183AABB	ETX	BCC
ASCII value	H06	H3130	H59(79)	H3039	H3031	H3034	H3931383341414242	H03	

② In case that the register Format of register No. is the continuous reading of direct

Format name	Header	Station No.	Command	Registration No.	Number of data	Data	Tail	Frame check
Frame (Example)	ACK	H10	Y(y)	H09	H04	H9183AABB	ETX	BCC
ASCII value	H06	H3130	H59(79)	H3039	H3034	H3931383341414242	H03	

(4) Response Format (NAK response)

Format name	Header	Station No.	Command	Registration No.	Error code (Hex 2Byte)	Tail	Frame check
Frame (Example)	NAK	H10	Y(y)	H09	H1132	ETX	BCC
ASCII value	H15	H3130	H59(79)	H3039	H31313332	H03	

Item	Explanation
BCC	When command is one of lower case(y), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC, and sent.
Error code	Hex and 2 bytes (ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes that registered variable No. 1 of station NO. 1 is read. and BCC value is checked. And it is supposed that direct variable %MWO is registered and the number of blocks is 1.

① Computer request Format (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Registration No.	Tail	Frame check
Frame (Example)	ENQ	H01	y	H01	EOT	BCC
ASCII value	H05	H3031	H79	H3031	H04	

② For ACK response after execution of command (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Registration No.	Number of Blocks	Number of data	Data	Tail	Frame check
Frame (Example)	ACK	H01	y	H01	H01	H04	H23422339	ETX	BCC
ASCII value	H06	H3031	H79	H3031	H3031	H3034	H3233343232 333339	H03	

③ For NAK response after execution of command (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Registration No.	Error code	Tail	Frame check
Frame (Example)	NAK	H01	y	H01	Error code (2)	ETX	BCC
ASCII value	H15	H3031	H79	H3031	Error code (4)	H03	

7) Reading PLC Status (RST)

(1) Introduction

This is a function that reads flag list including operating status of PLC and error information.

(2) PC request Format

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Example)	ENQ	H0A	R(r)	ST	EOT	BCC
ASCII value	H05	H3041	H52(72)	H5354	H04	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII and added to BCC.

(3) Response Format (ACK response)

Format name	Header	Station No.	Command	Command type	PLC status data (Hex 20 Byte)	Tail	Frame check
Frame (Example)	ACK	H0A	R(r)	ST	Status data Format	ETX	BCC
ASCII value	H06	H3041	H52(72)	H5354	[ ]	H03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII and added to BCC, and sent.

<table>
<tr><th>Item</th><th>Explanation</th></tr>
<tr><td>PLC status data</td><td>PLC status data: data Format is 20 bytes in hex Format and converted into ASCII code. Its contents are constituted as below table after converting ASCII code into hex data.
; Status data Format
<table>
<tr><th>Data type</th><th>Flag name</th><th>Status data order
(Hex data)</th></tr>
<tr><td>UINT</td><td>PC_DEVICE_IDENTIFIER;</td><td>H00(L) ~ H01(H)</td></tr>
<tr><td>Byte</td><td>Logical;</td><td>H02(Offset)</td></tr>
<tr><td>Byte</td><td>Physical;</td><td>H03</td></tr>
<tr><td>Byte</td><td>_CPU_TYPE;</td><td>H04</td></tr>
<tr><td>Byte</td><td>_VER_NUM;</td><td>H05</td></tr>
<tr><td>WORD</td><td>_SYS_STATE;</td><td>H06(L) ~ H07(H)</td></tr>
<tr><td>Byte</td><td>_PADT_CNF;</td><td>H08</td></tr>
<tr><td>Byte</td><td>_Domain_ST;</td><td>H09</td></tr>
<tr><td>WORD</td><td>_CNF_ER;</td><td>H0A(L) ~ H0B(H)</td></tr>
<tr><td>WORD</td><td>_CNF_WR;</td><td>H0C(L) ~ H0D(H)</td></tr>
<tr><td>WORD</td><td>Reserved</td><td>H0E(L) ~ H0F(H)</td></tr>
<tr><td>WORD</td><td>Reserved</td><td>H10(L) ~ H11(H)</td></tr>
<tr><td>WORD</td><td>Reserved</td><td>H12(L) ~ H13(H)</td></tr>
</table></td></tr>
</table>

Remark

1) For the details of each flag, refer to GM7 User's Manual "Appendix 2, list of flag".

2) PC_DEVICE_IDENTIFIER, Logical, and Physical are dedicated to be used only for system, that it should not be processed.

(4) Response Format (NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Byte)	Tail	Frame check
Frame (Example)	NAK	H0A	R(r)	ST	H1132	ETX	BCC
ASCII value	15	3041	5272	5354	31313332	03	

Item	Explanation
BCC	When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII and added to BCC, and sent.
Error code	Hex and 2 bytes(ASCII code, 4 bytes) indicate error type. For the details, see 8.1.8 Error codes.

(5) Example

This example supposes that the status of GM7 Base Unit of station NO. 1 is read.

¯Computer request Format (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Example)	ENQ	H01	R(r)	ST	EOT	BCC
ASCII value	H05	H3031	H52(72)	H5354	H04	

α For ACK response after execution of command (PC GM7 Base Unit)

Format name	Header	Station No.	Command	Command type	Status data	Tail	Frame check
Frame (Example)	ACK	H01	R(r)	ST	See status data Format	ETX	BCC
ASCII value	H06	H3031	H52(72)	H5354		H03	

β For NAK response after execution of command (PC GM7 Base Unit)

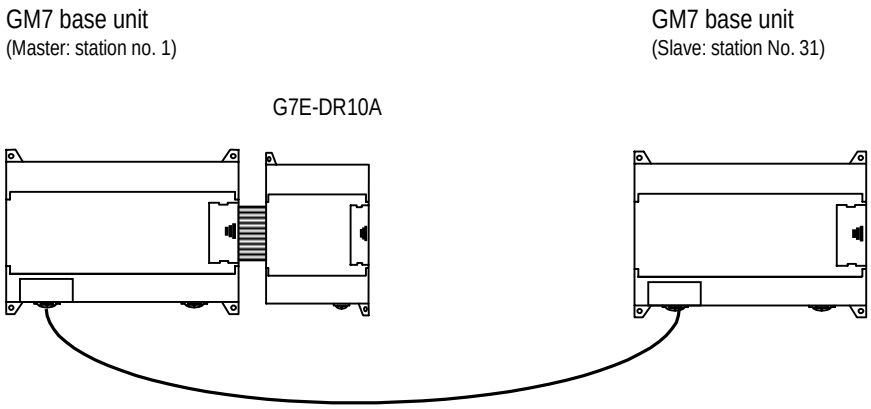
Format name	Header	Station No.	Command	Command type	Error code	Tail	Frame check
Frame (Example)	NAK	H01	R(r)	ST	Error code (2)	ETX	BCC
ASCII value	H15	H3031	H52(72)	H5354	Error code (4)	H03	

8.1.7 1:1 Built-in communication between GM7's

1) Introduction

1:1 built-in communication between GM7's is that which constitutes a built-in communication system with the method of 1(master):1(slave). Setting Base parameter and communication parameter in GMWIN can easily constitute this system. Communication protocol currently applied is the same with Cnet I/F used for G SERIES. Main functions are following.

- It can organize input(I), output(Q), and internal memory(M) area into 64 data access blocks by WORD unit,
  and set a communication time-out limit for each block.
- It can reestablish flag in relation with error codes and slave PLC operating mode according to parameter setting.
- It can reset flag related with error codes and sending/receiving error frequency of each parameter.
- It monitors communication status, using monitoring function of GMWIN.



1:1 built-in communication between GM7's

This communication cabling map is the same for (3) 1:1 Connecting between GM7's in 8.1.2 "System configuration method using built-in communication."

2) Parameter setting

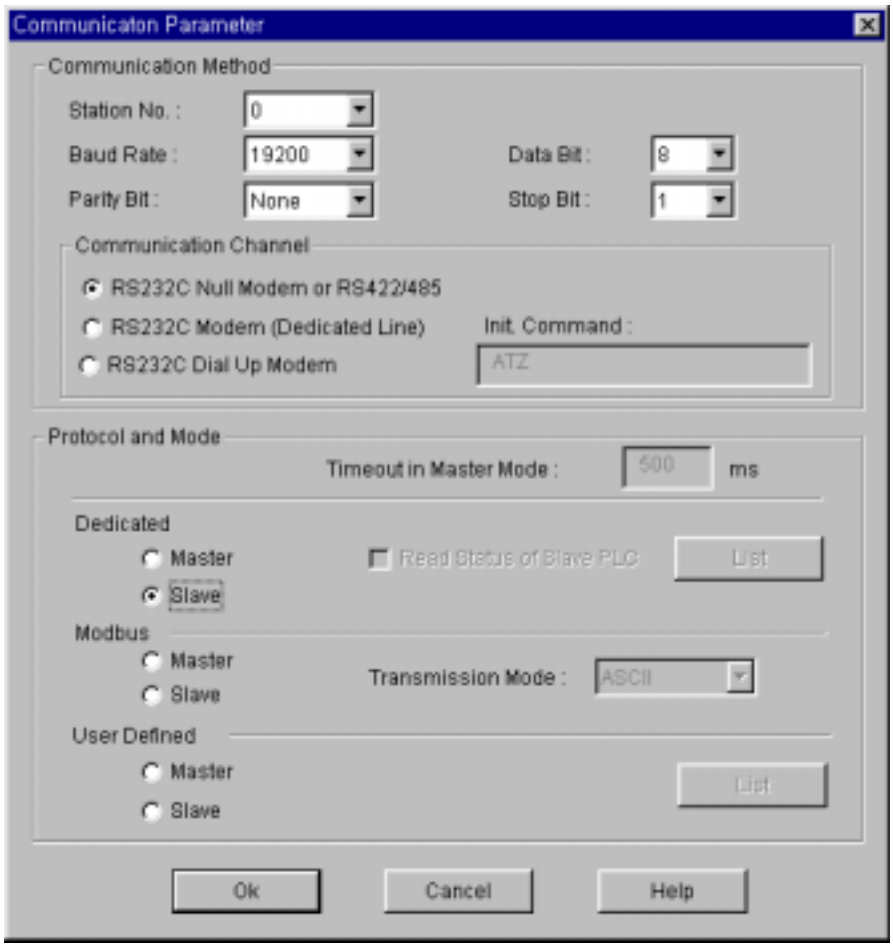
(1) Communication Parameter Setting

◆ Open a new project file from GMWIN

-GM7 must be selected as PLC type.

-Open a new project file for each of the master and the slave.

◆ After selecting communication parameter from GMWIN and clicking twice, this window comes up.

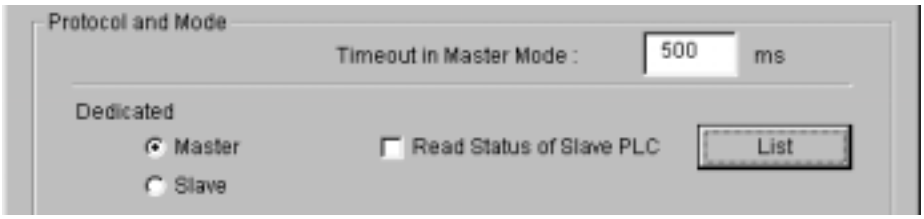


◆ Set according to the following table

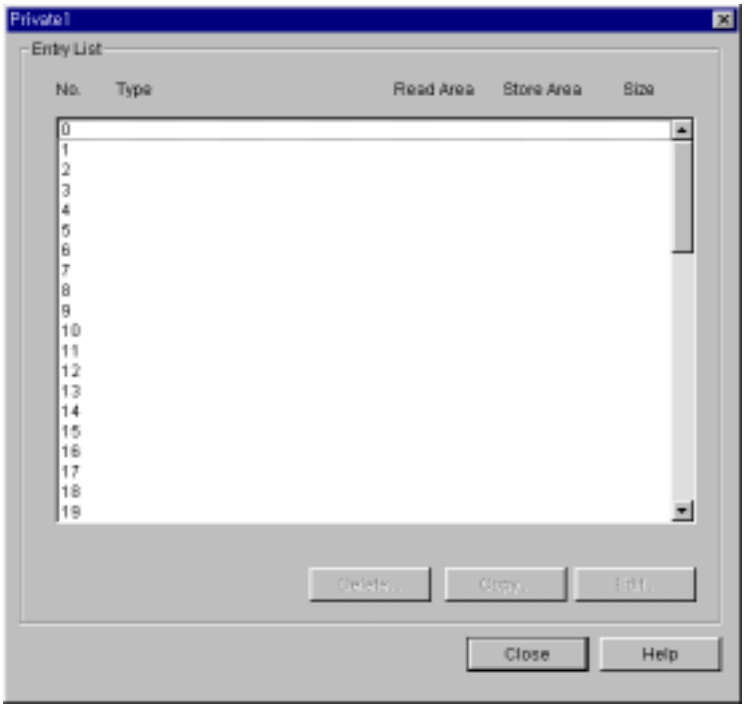
Item	Contents
Station No.	Sets one of station from 0 to 31.
Baud rate	Sets one of 1200, 2400, 4800, 9600, 19200, 38400, 57600 bps
Data bit	Sets one of 7 or 8 Bits
Parity bit	Sets one of none, Even, Odd
Stop bit	Sets one of 1 or 2 Bit(s)
Communication channel	• RS232C null modem or RS422/485: can be selected as a communication channel when communication is processed by built-in functions of GM7 Base Unit or Cnet I/F module (G7L-CUEC). • RS232C dedicated modem: can be selected when communication is processed by Cnet I/F module (G7L-CUEC). • RS232C dial-up modem: can be selected when common modem communication calling the opponent station is processed by Cnet I/F module (G7L-CUEC). * Notes: RS232C dedicated modem and RS232C dial-up modem communication can be processed only by Cnet I/F module (G7L-CUEC) supporting RS-232C, not Cnet I/F module (G7L-CUEC) supporting RS-422/485.
Timeout in Master Mode	• It's an interval waiting after sending request frame from Master GM7 before receiving a response. • default value is 500ms. • Setting must be done in consideration of maximum interval of sending and receiving cycle of a master PLC. • If the time out is less than the maximum interval of the s/r cycle, error can occur.
Dedicated Master/Slave	Master GM7 can read from and write on Slave GM7.
Read status of slave PLC	can be select especially when you read Slave GM7 for monitoring, but not for the other purposes, lest it may cause decreasing communication speed.

(2) Setting registration list

◆ If you click 'master' from 'exclusive use' in 'protocol and sending mode,' 'registration list' button will be activated.



◆ Click the button to open the registration list window.



(3) Total 64 data blocks can be assigned. But it's not possible to set a register number.

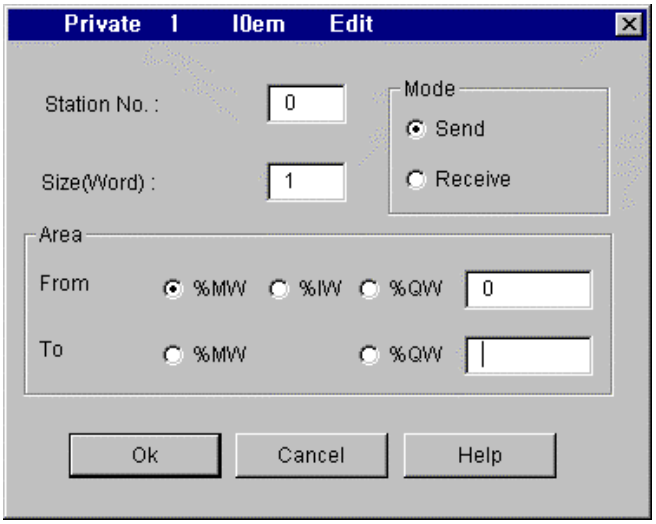
(4) Sending and receiving data size can be set up to 60 WORDs. There's no cycle for sending and receiving.

◆ Setting area

-Sending: reading area I/Q/M, saving area Q/M

-Receiving: reading area I/Q/M, saving area Q/M

(6) This is a window you can change 'exclusive use 1' setting.



◆ Station number: set the number of the slave or opponent station.

◆ Mode: click 'send' for writing data on the slave station, or 'receive' for reading from it.

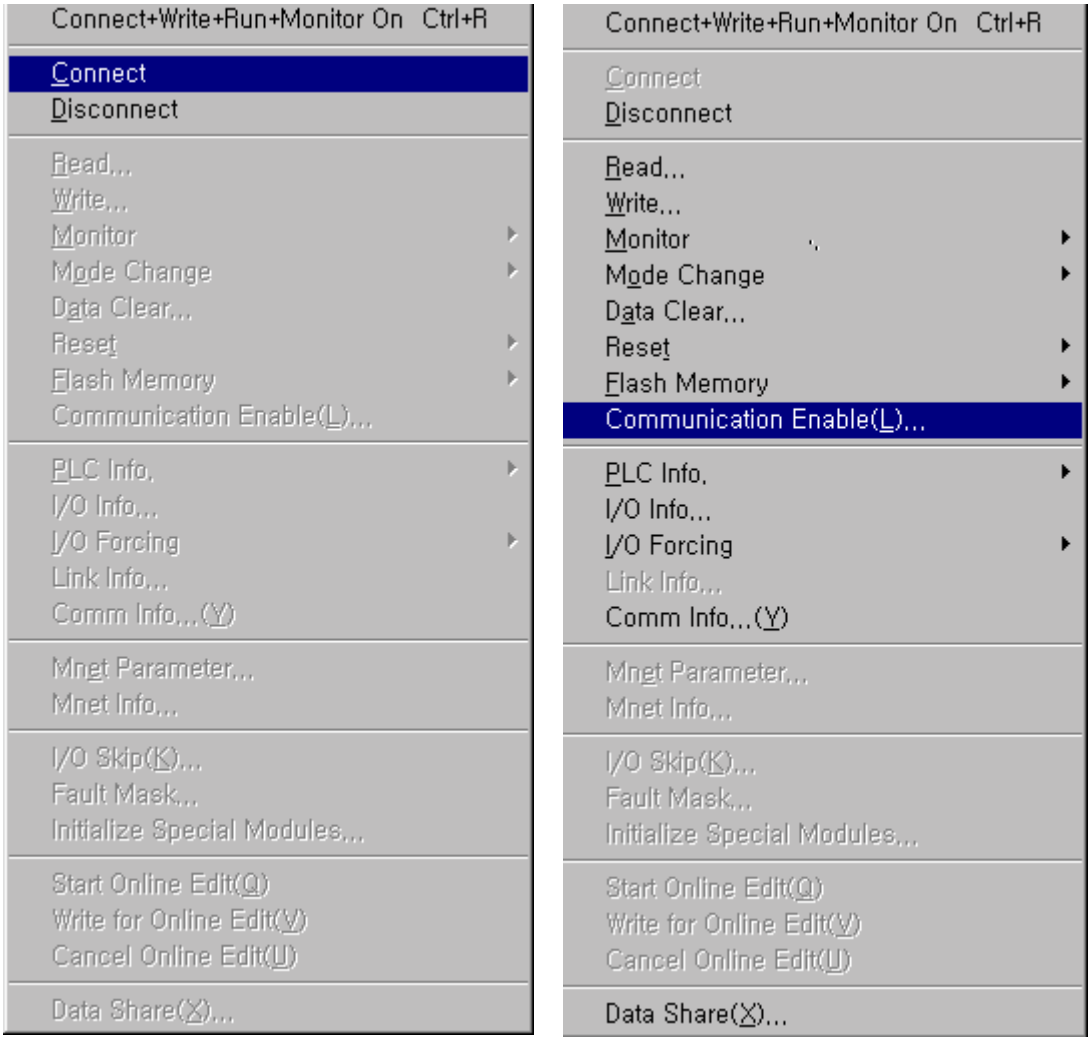
◆ Size: data size for reading and writing of the master station can be specified up to 60 WORDs. A WORD is a unit.

◆ Area:

Item	Send mode	Receive mode	Indication
Area(from)	that is in the master station to temporarily save the data to be written.	that is in the slave station for the data to be read.	When selecting %MW0, click '%MW' and enter '0' in the blank next to it.
Area(to)	that is in the slave station to write the data.	that is in the master station to temporarily save the data to be read.	when selecting %QW0.1.0, click '%QW' and enter '0.1.0' in the blank next to it.

3) Setting Communication Enable

To process 1:1 built-in communication between GM7's, after setting communication parameter and constituting program, you must access the master GM7 through GMWIN, click 'connect(C)' of 'online (O)' in menu bar, and set 'Communication Enable(L)' of 'online (O)' in the same menu bar.



4) Flag related with operating status

(1) Sending/receiving error count for each station (total 32 stations)

◆ Flag name: _MRS_ERR_CNT [n] (n=0-31)

◆ Data type: array byte

◆ Contents: each station can renew number of errors. Namely, the number of s/r errors of the station no. 1 is renewed at _MRS_ERR_CNT [0], and the number of the station no. 31, at _MRS_ERR_CNT [31].

(2) Sending/receiving error contents of each station (total 32 stations)

◆ Flag name: _MRS_ERR [n] (n=0-31)

◆ Data type: array byte

◆ Contents:

-Error code 1: error of time overrun responding time for s/r.

-Error code 2: error of NAK time

(3) Slave PLC mode and error contents of each station (total 32 stations)

◆ Flag name: _SRS_STATE [n] (n=0-31)

◆ Data type: array byte

◆ Contents:

-0 bit: error status of the slave PLC (1: error, 0: normal)

-1-3 bit: reserved

-4-7 bit: operating mode of the slave PLC

4 Bits: STOP/ 5 Bits: RUN/ 6 Bits: PAUSE/ 7 Bits: DEBUG

(4) Status flag of the master PLC

◆ Flag name: _MRS_STATE [n] (n=0-31)

◆ Data type: array byte

◆ Contents:

-2 Bit: overextending M area when setting communication parameter

(5) Max/min/current sending/receiving cycle of set parameter

◆ Flag name:

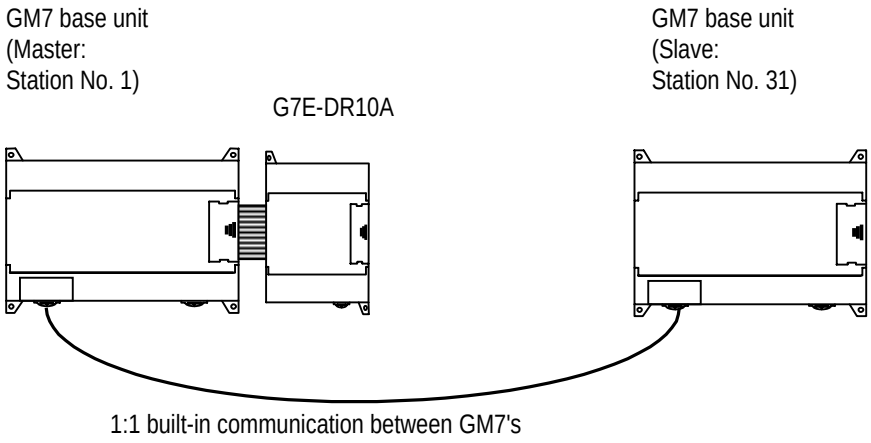
-(Time Type) _MRS_SCAN_MAX

-(Time Type) _MRS_SCAN_MIN

-(Time Type) _MRS_SCAN_CUR

◆ Contents: the interval between after sending and before receiving

5) Example



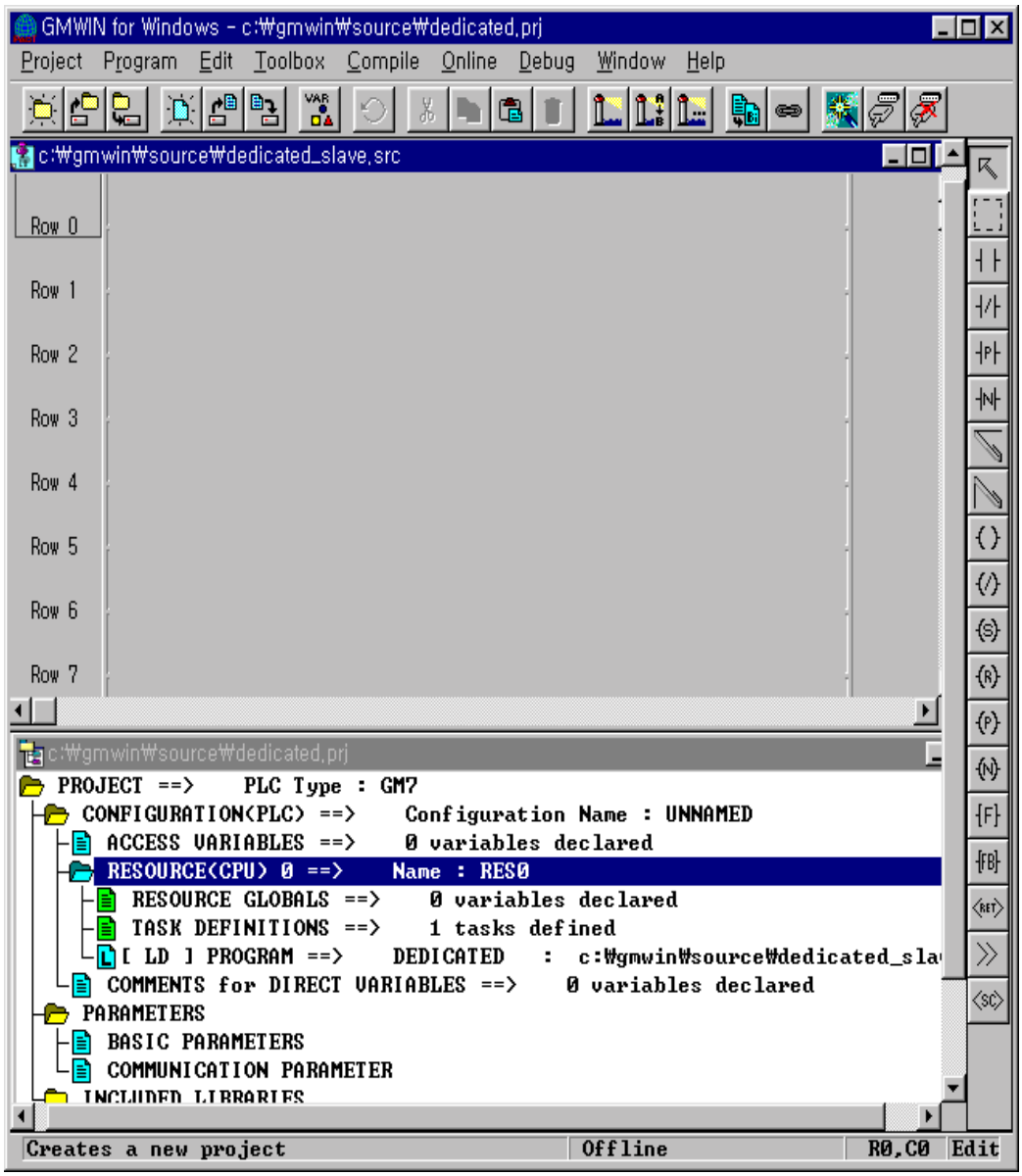
The following example uses the above diagram to explain the operation of GM7 Base Unit.

-The data of the master GM7 Base Unit is converted by ROL F/B and MOV F/B and sent to be written on the output contact point of the slave GM7 Base Unit. And in return, the master GM7 Base Unit reads the data that is written on the output contact point of the slave GM7 to write them on the output contact point of extended digital input/output module, G7E-DR10A.

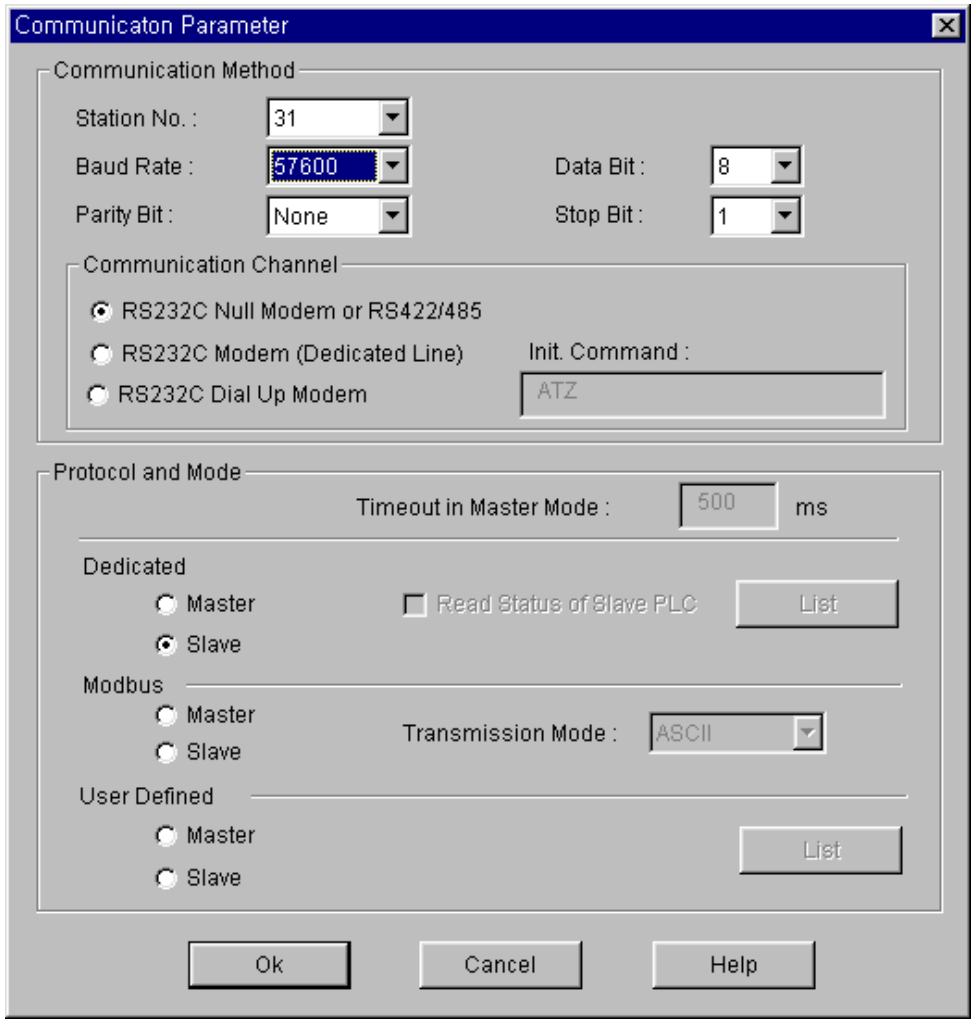
(1) Setting communication parameter of the slave station and its program

Work on the slave station 31.

Open a new project file and a new program for the slave station.



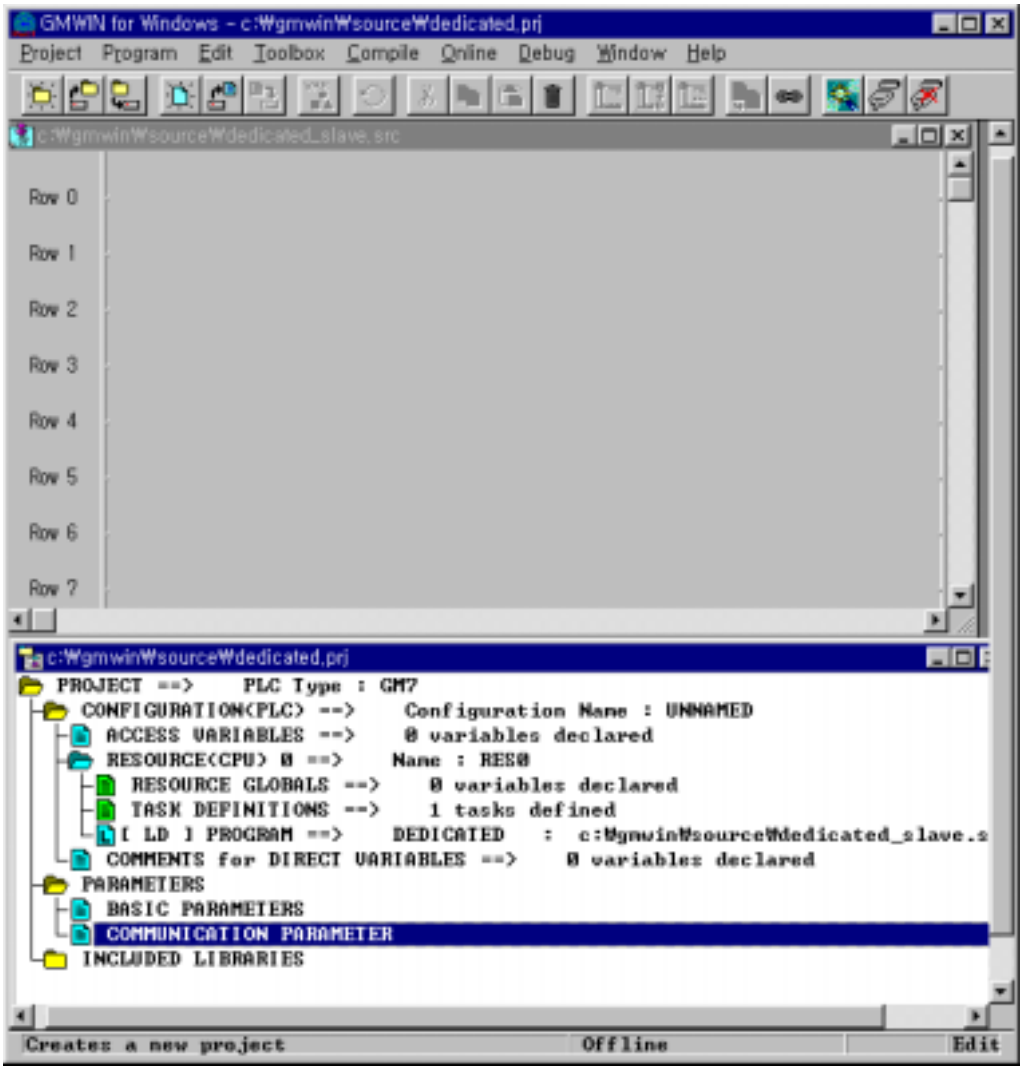
If you double click the 'communication parameter' window in GMWIN, you can see the following window of the 'communication parameter.'



-Set parameters as the following table.

Communication Method						Protocol and mode
Station no.	Baud rate	Data bit	Parity bit	Stop bit	Communication channel	Dedicated
31	57600	8	None	1	RS232C null modem or RS422/485	Slave

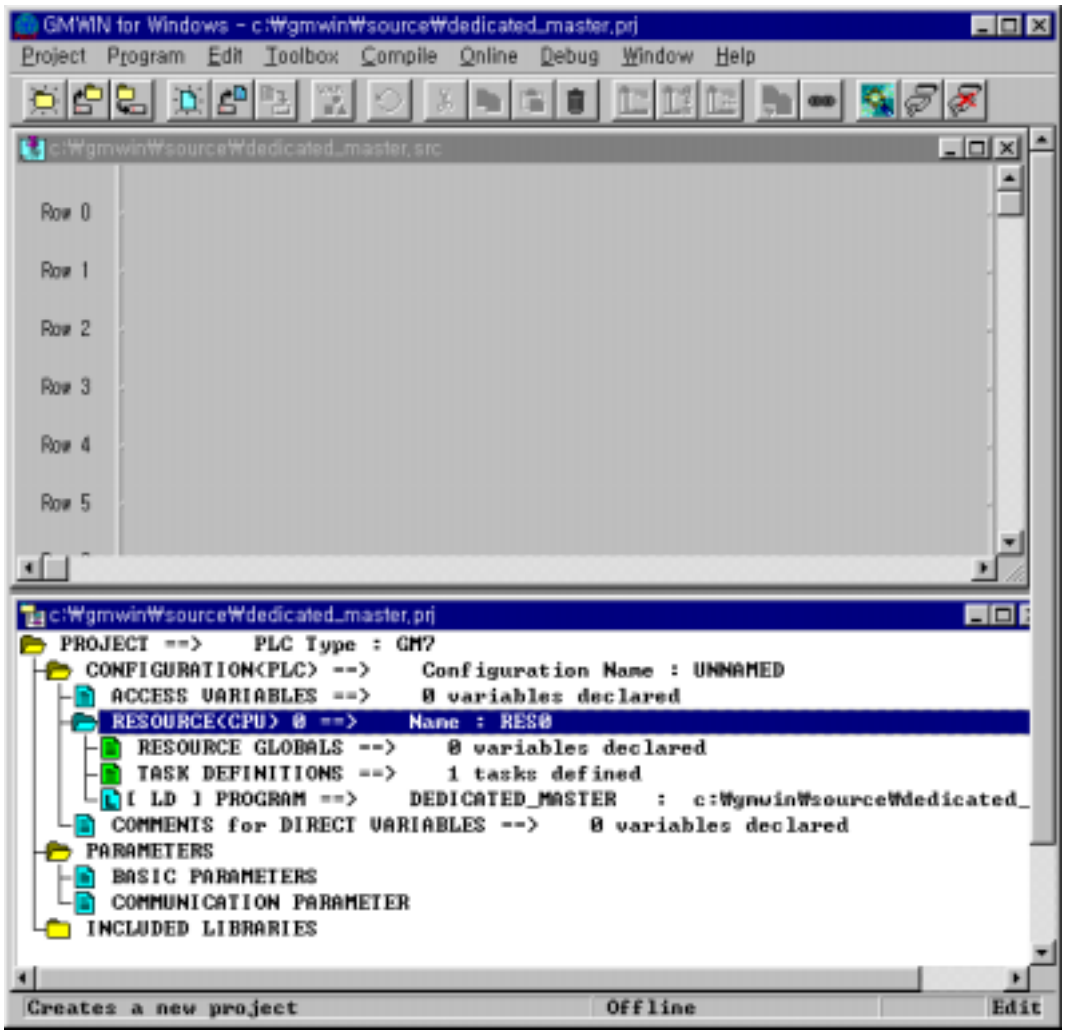
Program like the following screen and download to the slave GM7 Base unit. For programming and downloading, refer to GMWIN User's Manual.



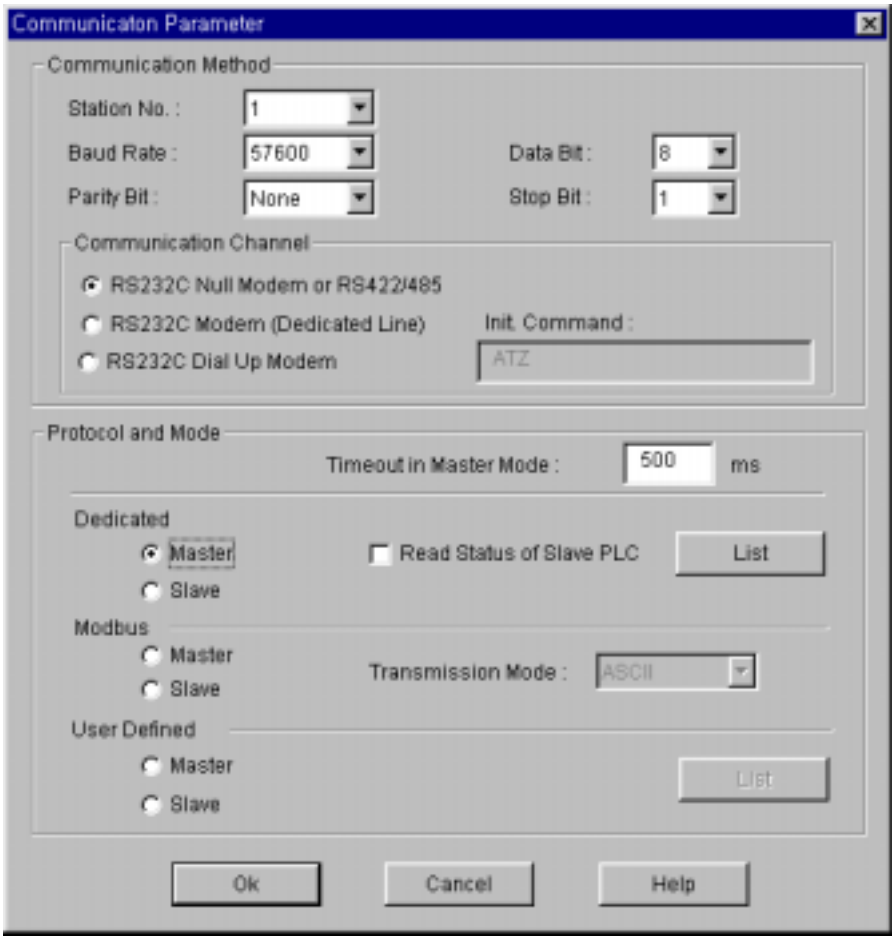
(2) Setting communication parameters for the master station and programming

1 Work at the master station no. 1

2 Open a new project file and a new program for the master station.



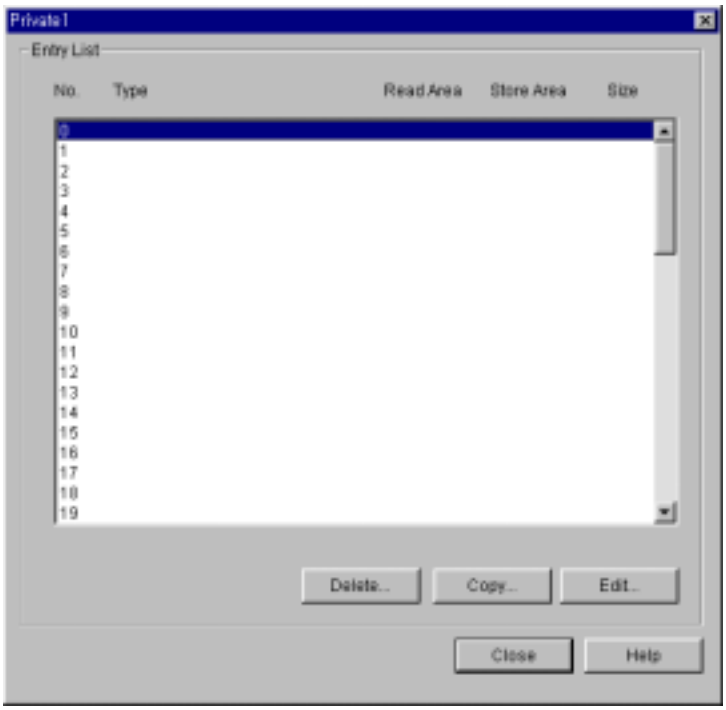
3 Select a communication parameter from the GMWIN parameter and double-click to open the communication parameter menu window.



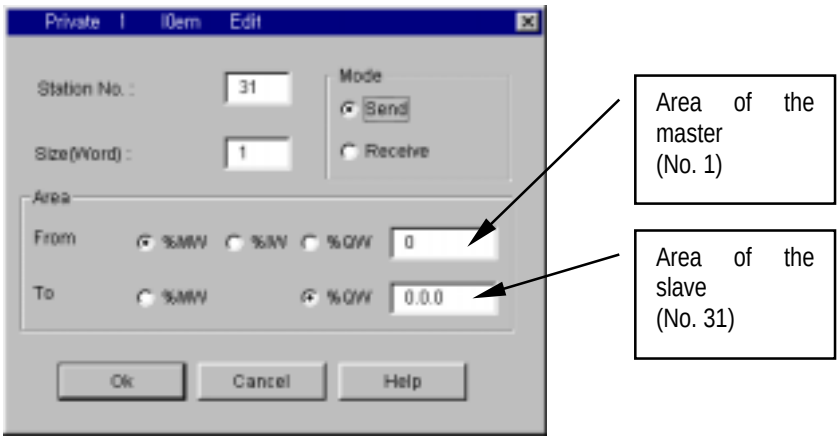
- Set parameters like the following table and click 'OK' button.

Communication method						Protocol and mode	
Station no.	Baud rate	Data bit	Parity bit	Stop bit	Communication channel	Dedicated	Read status of slave
1	57600	8	None	1	RS232C null modem or RS422/485	Master	Not selected

4 Click 'List' button to activate registration list window.



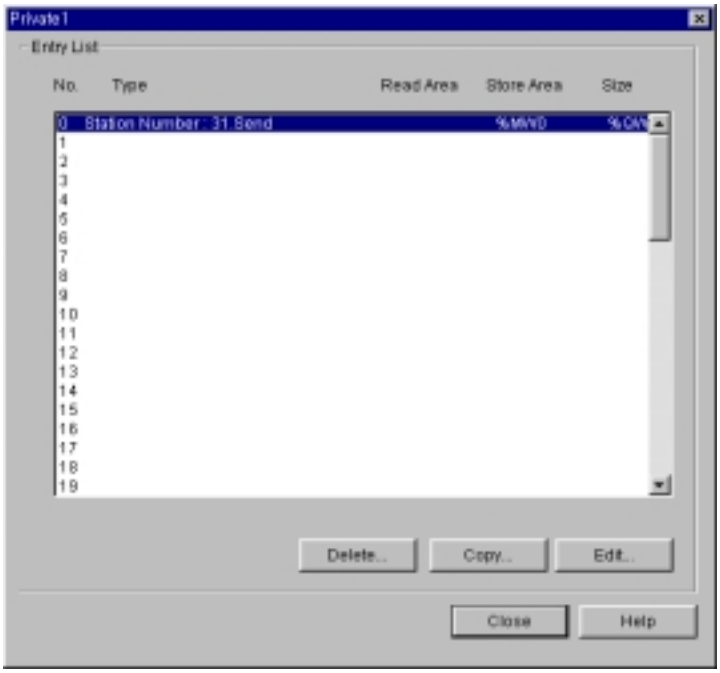
5 If the list number '0' in 'List' window is double clicked, another window for 'Private 1 item 0 edit' is open.



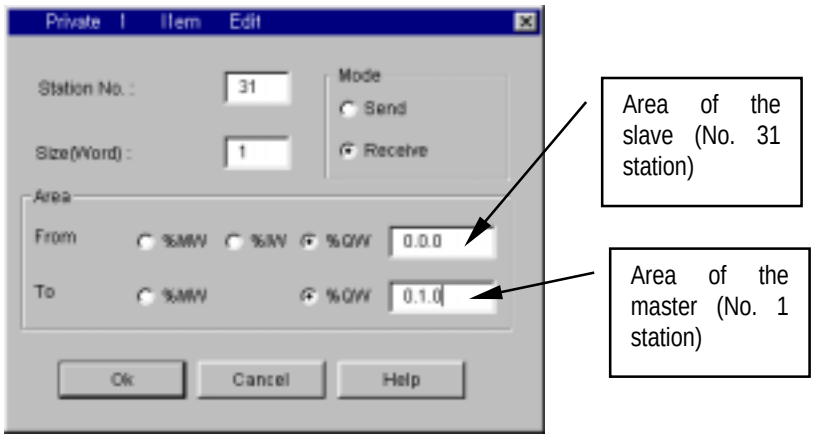
- - Set parameters like the following table and click 'OK' button.

Station No.	Size	Mode	Area to read(From)	Area to save(to)
31	1	Send	%MW0 (See the above)	%QW0.0.0 (See the above)

The registration list '0' registered in the registration list can be confirmed through a window like the following.



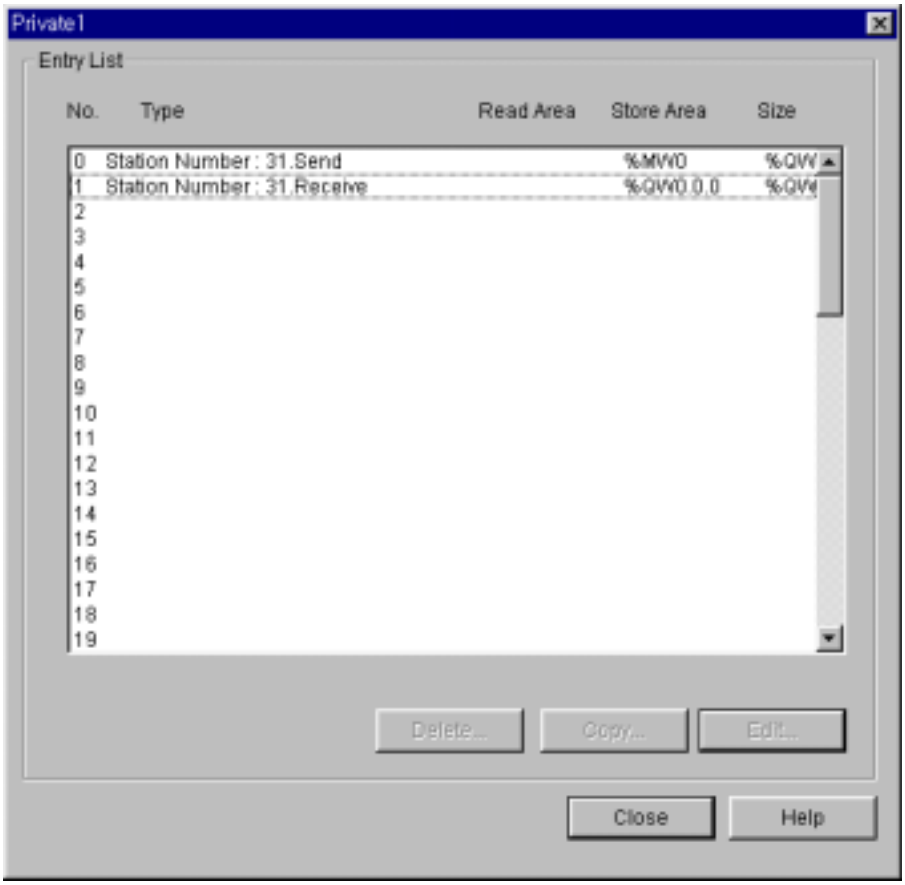
7 Double click 'entry list 1' to be registered like the following



- Set parameters like the following table and click 'OK' button.

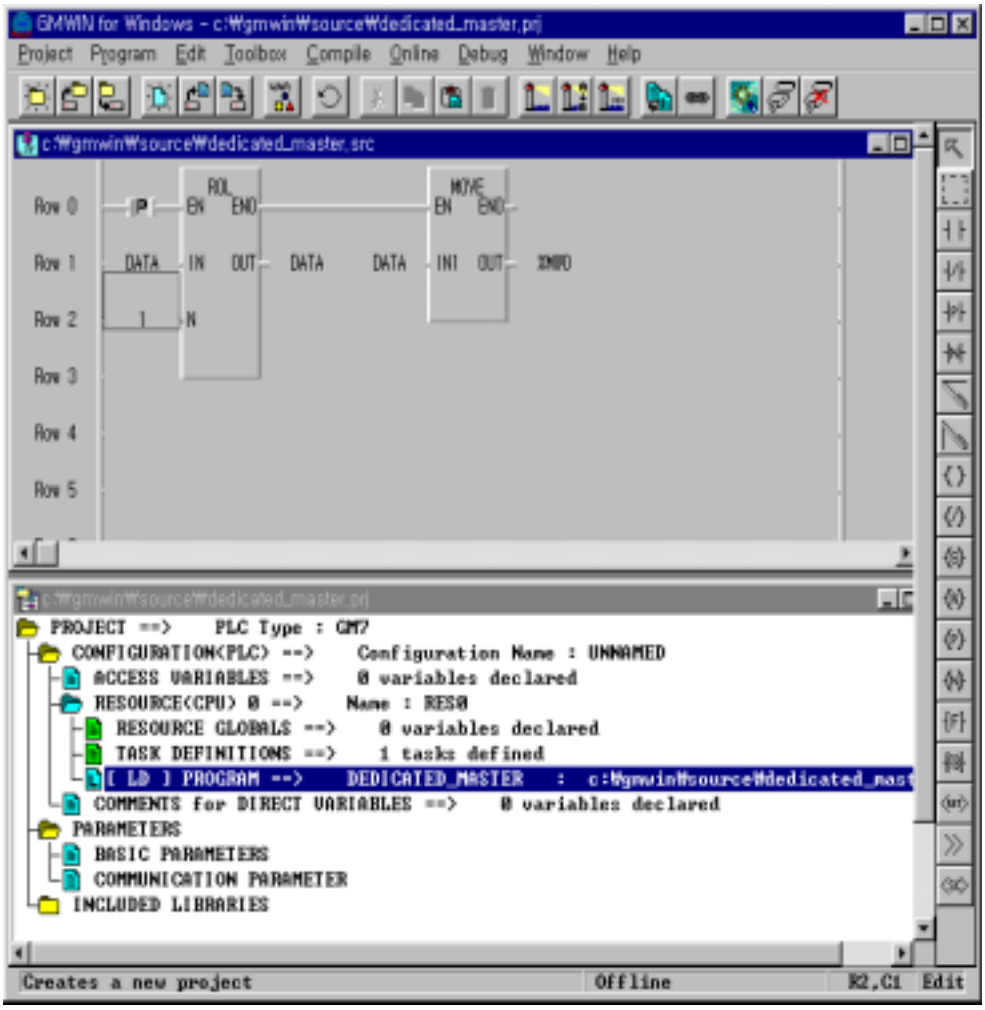
Station No.	Size	Mode	Area to read(From)	Area to save(To)
31	1	Receive	%QW0.O.O (See the above)	%QW0.1.0 (See the above)

8 Confirm the registration of the 'List' 0 and 1 through a window like the following and close this window to go back to communication parameter window.

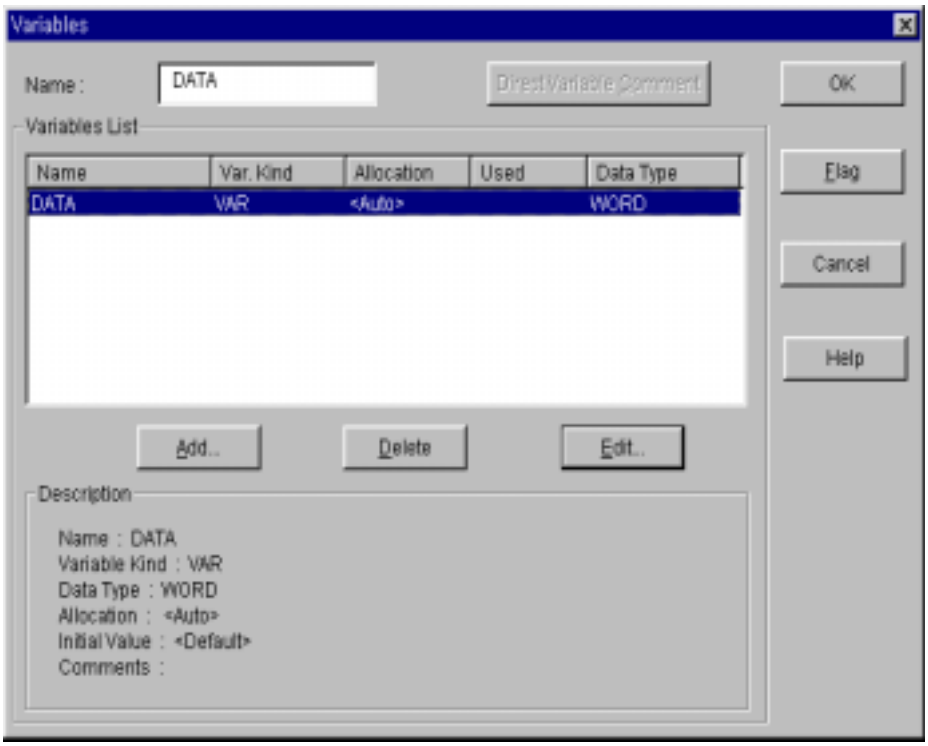


9 Close setting parameters by clicking 'Close' button in the communication parameter window.

10 Program like the following screen and download to the slave GM7 Base unit. For programming and downloading, refer to GMWIN User's Manual.



11 Variable of the program, 'DATA' is WORD type and its first value is H'FF.



12 Click 'Communication Enable' to start 1:1 built-in communication between GM7 Base Units on line.

13 Remove RS-232C communication cable for GMWIN and connect GM7 Base Units for 1:1 built-in communication between GM7 Base Units.

8.1.8 Error code

Error code	Error type	Error condition and causes	Treatment
H0001	PLC system error	* Interface with PLC is impossible.	* On/Off the power
H0011	Data error	* Errors occurred when exchanging ASCII data to numbers.	* Check if other letters than capitals/small letters, numbers, and ('%','_','.') in variable and data, correct and execute again.
H0021	Command error	* Set a wrong device memory that is to use commands other than w (W), r(R), x (X), y (Y), s (S)	* Check commands.
H0031	Command type error	* Wrong command type that is to use characters like wSS, wSB using other letters from "SS" or "SB"	* Check command type
H1132	Device memory error	* Set wrong device memory other than m (M), q (Q), I (I) at GM7.	* Check device type
H1232	Data size error	* The number of data in execution is 0 or bigger than 128 bytes.	* Correct length of data (If data type is bite, the number of data must be from 1 ~ 128.
H2432	Data type error	* When use other characters than x (X), b (B), w (W), d (D) at GM7. * When use b (B), d (D) at GM7. Ex1) Use commands like %db or %dd.	* Check data type and execute again.
H7132	Variable request Format error	* When omit %. * When omit '.' in Qx commands.	* Check Format, correct and execute again.
H2232	Area exceeding error	* When exceed assigned area. Ex1) %QX0.0.64 → Area exceeding error %MB0.0.8 → Area exceeding error Ex2) When an address isn't decimal, like %MX00A, %MB00A. Ex3) When set M area size as 2Kbytes in GMWIN, but exceed over 2 Kbytes like %MB400.	* Correct the size within the assigned area and execute again.

Error code	Error type	Error condition and causes	Treatment

H0190	Monitor execution error	* Exceeding limit of register No.	* Rearrange the monitor register no. not to go over than 9 and reset.
H0290	Monitor register error	* Exceeding limit of register No.	* Rearrange the monitor register no. not to go over than 9 and reset.
H6001	Syntax error	* When use commands that aren't supported. Ex1) When use variables like %MX100, %QX0.0.0 in RSB command	* Be familiar with the manual. * Check if the system stopped. * Off/on the power to restart.
H6010	Syntax error	* OVER-RUN, FRAME error	* Be familiar with the manual.
H6020	Syntax error	* TIME_OUT error	* Confirm the setting of the communication ports of RS-232C. * Off/on the power to restart.
H6030	Syntax error	* Syntax error in commands	* Check if each sends frame has ENQ, EOT.
H6040	Syntax error	When a FRAME text exceeds over 256 bytes.	* Rearrange send frame not to go over 256 bytes.
H6050	Syntax error	* BCC error	* Check if BCC is right.

8.2 User Defined Protocol Communication

8.2.1 Introduction

User Defined Protocol Communication allows users who do communication between GM7 Basic Unit and other kind of device to define the other company's protocol at G SERIES PLC. There're a number of kinds of protocols made by many companies, that it's difficult to have all protocols in it. So if a user defines a protocol that can be applied to his/her purpose, GM7 Basic Unit executes the communication with the other kind of device through the defined protocol. For this, protocol frame must be defined in GMWIN (Version 3.3 or higher). And exact knowledge about the contents of the protocol defined by the user is vital in making the communication possible. GMWIN (Version 3.3. or higher) can download a user defined protocol frame into GM7 Basic Unit and it is saved(it is not erased by power's off/on). But protocol frames are damaged to download with changes of parameter, or to fail to back up the data, caused by lower back-up battery voltage than the standard. For using user-defined mode, he/she should program with function block controlling sending/receiving of PLC as well as edit frames. This section explains UDPC setting & usage.

Remark

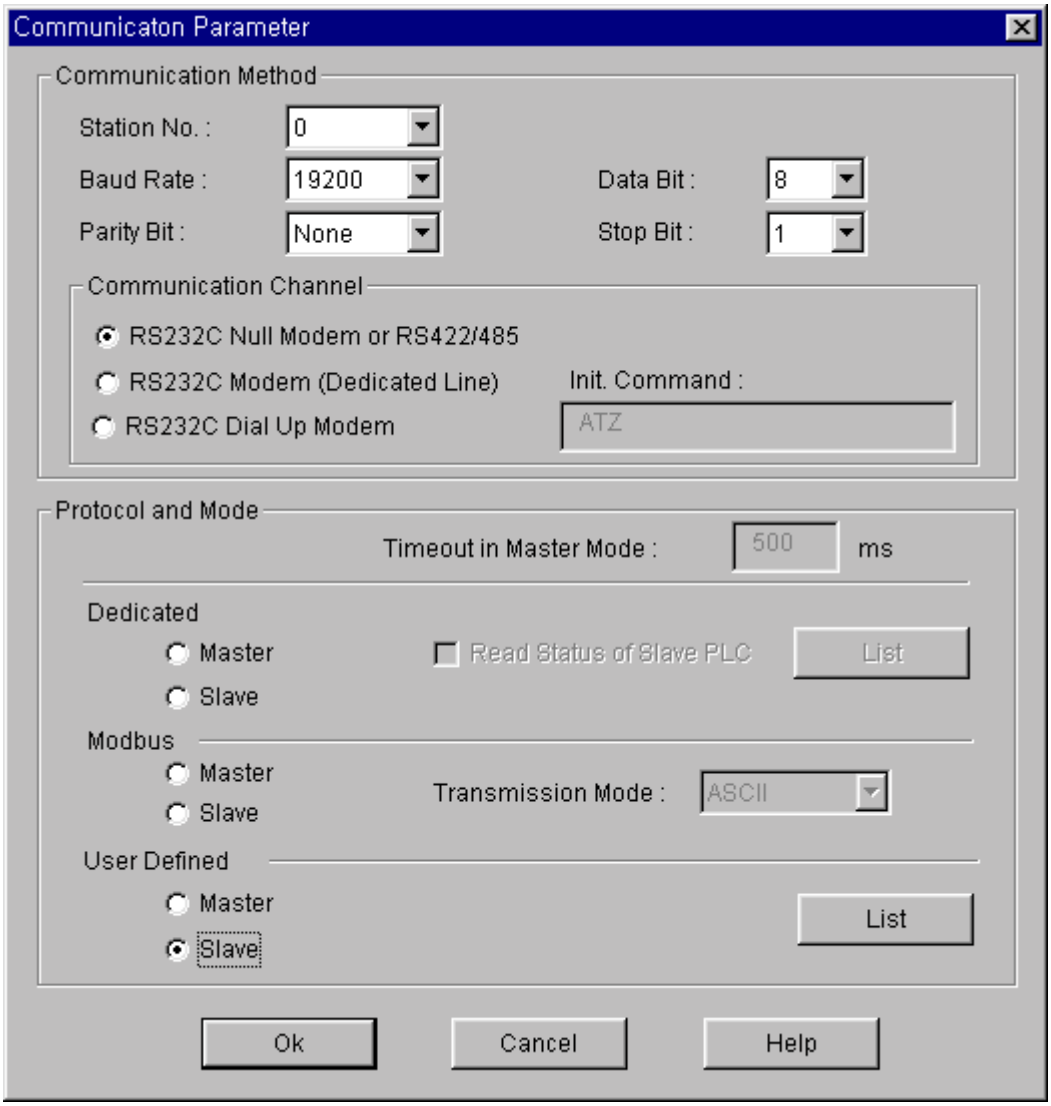
1) All numerical data can use hexadecimal, decimal, and binary type. If we convert decimal 7 and 10 into each type:
   Hexadecimal : H07, H0A or 16#07, 16#0A
   Decimal : 7, 10
   Binary : 2#0111, 2#1010

8.2.2 Parameter Setting

1) Setting Communications Parameter

(1) Open a new project file from GMWIN
1 Select GM7 as PLC type
2 Make a separate project file for each of the master and slave.

(2) After setting communication parameter at GMWIN. Double click it to activate this window.



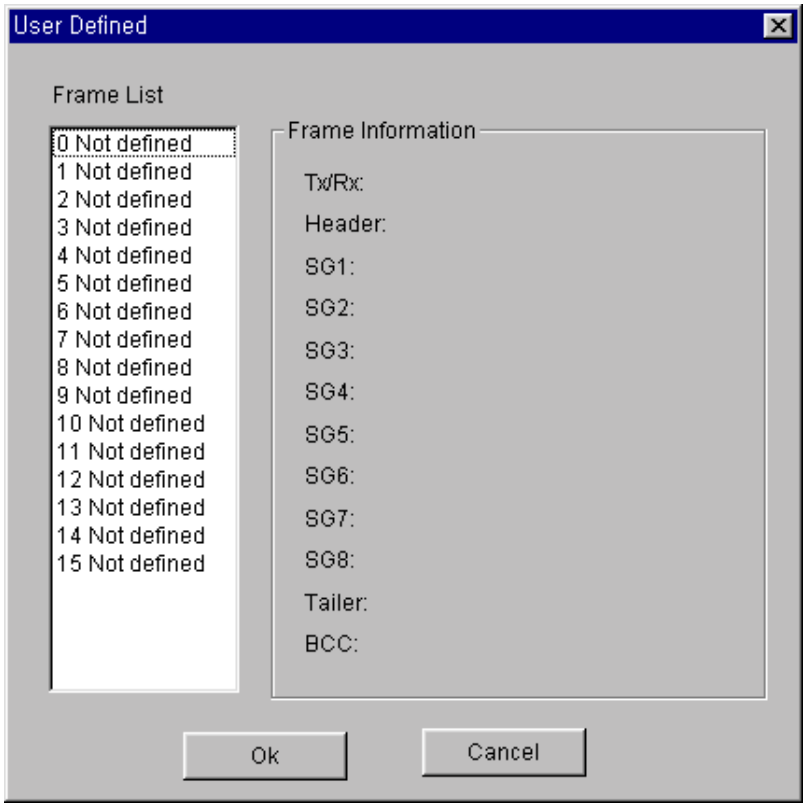
(3) Set according to the following table.

Item	Setting range
Station No.	Station no. from 0 to 31.
Baud Rate	1200, 2400, 4800, 9600, 19200, 38400, 57600 bps
Data Bit	7 or 8 bits
Parity Bit	0, Even or Odd
Stop Bit	1 or 2 bit(s)
Communication Channel	● RS232C Null Modem or RS422/485 : It's a communication channel for the communication, using GM7 base unit's built-in communication and Cnet I/F module (G7L-CUEC). ● RS232C Modem (Dedicated Line) : It's to be selected for the communication, using an dedicated modem with Cnet I/F module (G7L-CUEB). ● RS232C Dial Up Modem : It's to be selected for the general communication connecting through the telephone line by dial up modem and Cnet I/F module (G7L-CUEB). Footnote) Using Cnet I/F module (G7L-CUEB) supporting RS232C, RS232C dedicated or dial-up modem communication can be done, but not through Cnet I/F module (G7L-CUEC) supporting RS422/485.
Timeout in Master Mode	● It's the time waiting a responding frame since the master GM7 base unit sends a request frame. ● The default value is 500ms. ● It must be set in consideration of the max. periodical time for sending/receiving of the master PLC. ● If it's set smaller than the max. send/receive periodical time, it may cause communication error.
User Define Master / Slave	If it is set as the master, it's the subject in the communication system. If it's set as the slave, it only responds to the request frame of the master.

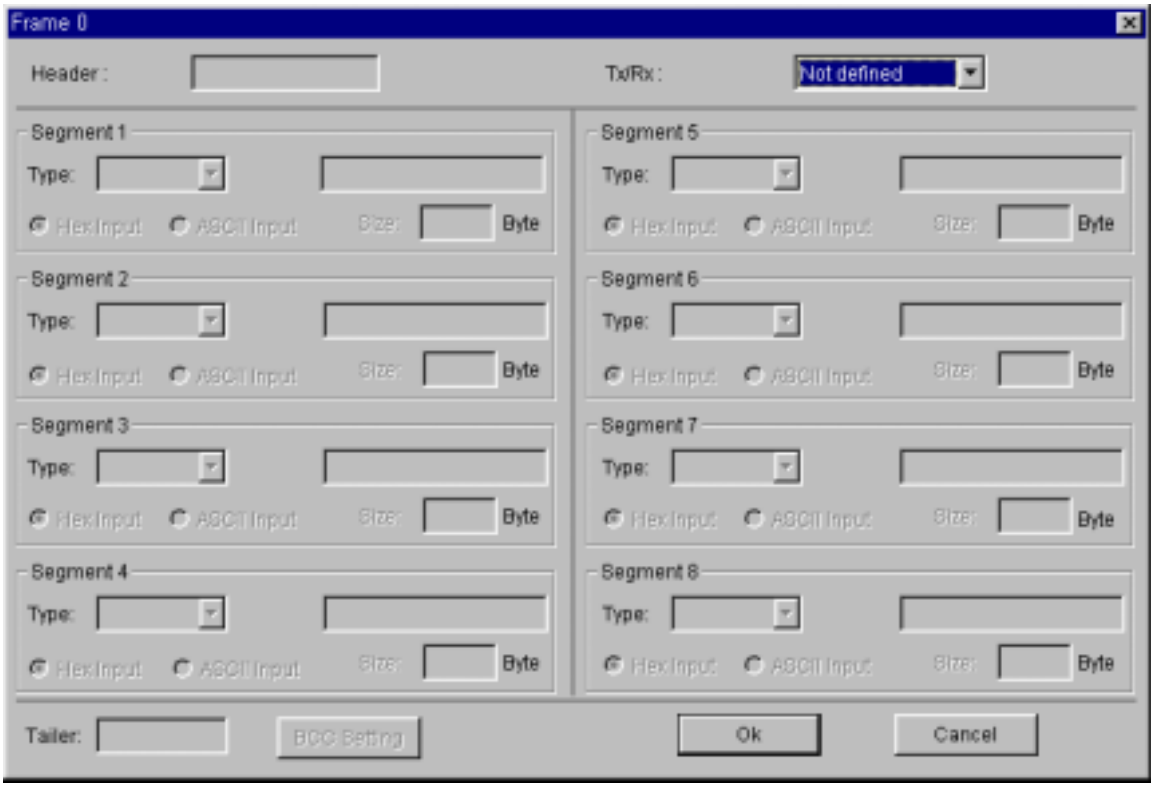
2) Setting frame

(1) Select one out of user defined terms of protocol and mode in communication parameter, registration "List" button is activated.

(2) Click “List” button to activate the following window.



(3) Select one of 1 15 in frame list to open the following window.



1 Frame specification

- Header
  - Used in [Header] type.
  - Possible characters as headers are 1 alphabet letter, 1 numeric number, or control characters like "NUL (H00)", "STX (H02)", "ETX(H03)", "EOT(H04)", "ACK(H06)", "NAK(H15)", "SOH(H01)", "ENQ(H05)", "BEL(H07)", "BS(H08)", "HT(H09)", "LF(H0A)", "VT(H0B)", "FF(H0C)", "CR(H0D)", "SO(H0E)", "S1(H0F)", "DLE(H10)", "DC1(H11)", "DC2(H12)", "DC3(H13)", "DC4(H14)", "SYN(H16)", "ETB(H17)", "CAN(H18)", "EM(H19)", "SUB(H1A)", "ESC(H1B)", "FS(H1C)", "GS(H1D)", "RS(H1E)", "US(H1F)" and "DEL(H7F)."

    Ex1) [NUL], [ENQ] ( )
    Ex2) NUL, ENQ (×: [ ] isn't used)

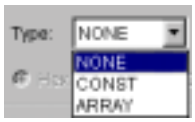
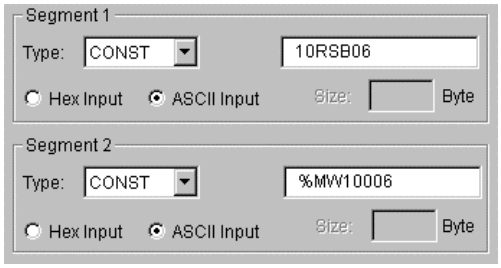
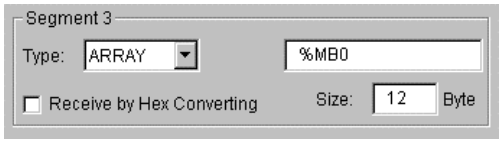
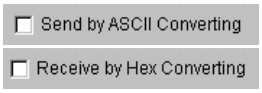
  - It is allowed to be only 3 consecutive characters.

    Ex1) [ENQ][STX][NUL] ( )
    Ex2) [A][NUL][ENQ][STX] (×: 4 consecutive characters are used)

- Tx/Rx
  - Not defined : It is the initial value that doesn't declare a frame format.
  - Send : It is that declares send frame.
  - Receive : It is that declares receive frame.

  - When Frame 0 window is activated, Tx/Rx term is set as "Not defined," and all the segments are not in activation.

- Segment (1-8): Enter segment by segment to separate fixed sending data area (CONSTANT) and variable data area (Array).

Item	Contents
	To set a segment type, there're NONE (not defined), CONST (fixed data area), ARRAY (variable data area). CONST declares commands and fixed data that are used for communication frame and ARRAY is used to input and save the data needed for interactive communication. ARRAY type must be always set by byte. Ex1) %MB0, %QB0.0.0 ( ) Ex2) %MX0, %MW0, %MD0, %QX0.0.0, %QW0.0. (×)
	This field is to declare commands and fixed data that will be used in communication frame and constant data to be declared by inputting. ASCII input must be done within 10 characters and hex within 20 characters. If the number exceeds the limit, set the next segment as the same type and continue to input there. Ex1) 10RSB06%MW10006  If the segment is declared as ARRAY type, although word type data is declared in CONST type, the related area is to be set by byte. Ex2) As an dedicated protocol communication, 10RSB06%MW10006 is a frame to execute reading 6 word data from %MW100 at the slave station no. 16. At the moment, ARRAY must be set in 6 words that is 12 bytes, as the area to save the data that is read. 
	It is a radio button to select the input type of commands. There're 2 kinds as hex or ASCII value. Ex1) ASCII : 1 0 R S B 0 6 % M W 1 0 0 Ex2) Hex : 31 30 52 53 42 30 36 25 57 44 31 30 30
	If ARRAY (variable data area) is set, it asks whether it would convert data to ASCII to send (at send frame), or convert to hexadecimal to receive(at receive frame).
	If ARRAY is set, the size of area is to be set by byte. **The unit is a byte.**

- Tail
  - Used in [Tail] type.
  - Possible characters as tails are 1 alphabet letter, 1 numeric number or control characters like “NUL (H00)”, “STX (H02)”, “ETX (H03)”, “EOT (H04)”, “ACK (H06)”, “NAK (H15)”, “SOH (H01)”, “ENQ (H05)”, “BEL (H07)”, “BS (H08)”, “HT (H09)”, “LF (H0A)”, “VT (H0B)”, “FF (H0C)”, “CR (H0D)”, “SO (H0E)”, “S1 (H0F)”, “DLE (H10)”, “DC1 (H11)”, “DC2 (H12)”, “DC3 (H13)”, “DC4 (H14)”, “SYN (H16)”, “ETB (H17)”, “CAN (H18)”, “EM (H19)”, “SUB (H1A)”, “ESC (H1B)”, “FS (H1C)”, “GS (H1D)”, “RS (H1E)”, “US (H1F)”, “DEL (H7F)”, and “BCC(HFE).”

    Ex1) [NUL], [EOT] ( )
    Ex2) NUL, EOT (×: [ ] isn’t used)

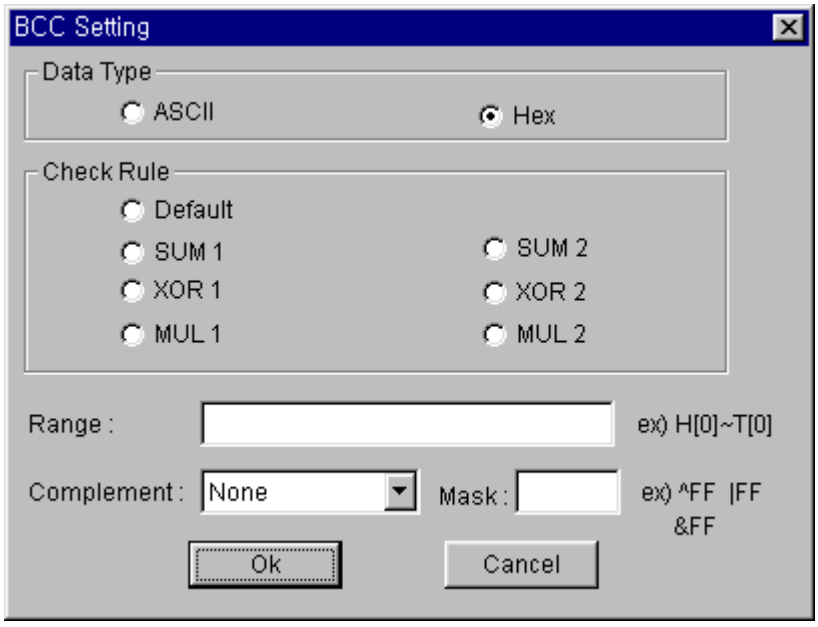
  - It is allowed to be only 3 consecutive characters.

    Ex3) [ENQ][STX][NUL] ( )
    Ex4) [A][NUL][ENQ][STX] (×: 4 consecutive characters are used)

  - It’s possible to use BCC that can detect errors. BCC must be set as [BCC] to be used. To set BCC contents, click “BCC Setting” button on the right side.

    Ex5)

- BCC setting: set BCC when it is needed.



Item		Contents
Data Type		ASCII adds 2 bytes BCC value in ASCII type to frame. Hex adds 1 byte BCC value in Hex type to frame. For the detailed setting BCC, refer to 8.1.6 “Execution of Commands”.
Check Rule	Default	It is that sum all the data from 2nd data to the data before the data marked as [BCC] and input the result to the [BCC] area
	SUM 1	BCC method uses sum like defaults, but the user can set the BCC area.
	SUM 2	BCC method is the same with SUM 1, but it’s used when the user masks any value to the last BCC value.
	XOR 1	BCC method is OR (EXCLUSIVE OR).
	XOR 2	BCC method is the same with XOR 1, but it’s used when the user masks any value to the last BCC value.
	MUL 1	BCC method is MULTIPLY that is, multiplication.
	MUL 2	BCC method is the same with MUL 1, but it’s used when the user masks any value to the last BCC value.
Range		H signifies header, S is for segment, and T is for tail. Ex1) When header is set as [ENQ][STX], tail is set as [EOT][ETX], and the range of setting BCC is to be from [STX] to [ETX], then set as **H [1]~T [1]**.
Complement		It is to set whether not to take complement number or to take the complement number of 1 or 2 at [BCC] value. If mask setting is done after taking a complement number, the user can set any value to do masking.
Mask		Sets any value and method of masking. Ex1) When masking by XOR method, using a value, HFF : ^FF Ex2) When masking by OR method, using a value, HFF : |FF When masking by AND method, using a value, HFF : &FF

Keys on Keyboard, for setting masking method

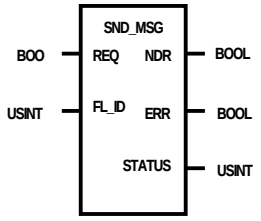
- Frame size
  - ASCII communication : max. 128 bytes
  - Hex communication : max. 256 bytes

- Flag (_RCV [n]: n is a frame list no.)
  - It’s a flag to indicate whether a user defined frame is received in the order set by the user.
  - It’s BOOL type and ARRAY type in the size of 6.
  - If the received frame is matched with the declared frame in frame list number 3, _RCV [3] starts blinking. (0 1 0)

- When frame receiving is done, GM7 base unit check if there’s any match between the received frame and the declared frame in frame list. If there is, let the _RCV [n] flag blink and save the received data in the assigned area.

8.2.3 Function Block

1) SND_MSG

Function block	Explanation
	**Input** REQ : Execute function block at rising edge(0 1) FL_ID : Frame list field number to send. **Output** NDR : When ends without error, this is set to 1 and keeps till the next request for function block. ERR : When an error occurs, this is set to 1 and keeps till the next request for function block. STATUS : When an error occurs, output error code.

(1) Function
This function block is to send the frame registered in the frame list.

(2) Error
Output error codes to the output STATUS. For the detailed information, refer to the error codes.

2) Error codes (Status of the function block)

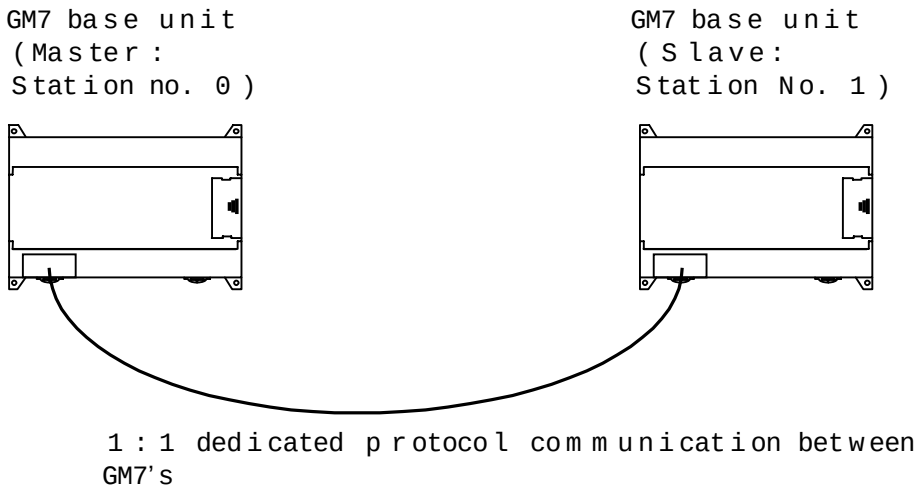
CODE	Error type	Meaning
06	Slave Device Busy	It's sending or waiting to receive
09	Parameter Error	Comm. Parameter setting, or communication enable setting error
10	Frame Type Error	The frame isn't for sending or set.

3) Error flag
Errors related with the frame list
Save in _UD_ERR [n].

Bit no.	Error type	Meaning	Remark
Bit 0	Frame Length Over	Error when a frame is bigger than 128 bytes (hexadecimal: 256 bytes).	Error flag on sending part
Bit 1	Device Area Over	Error when I/Q area (128 bytes), or M area is exceeded.	
Bit 2	BCC Setting Error	BCC setting error	
Bit 8	BCC Check Error	Error when received BCC value is wrong.	Error flag on receiving part
Bit 9	Mismatch Error	Error when a received frame doesn't match with the registered frame.	
Bit 10	Hex Change Error	Error when converting to hexadecimal.	
Bit 11	Frame List Error	Frame setting or communication enable setting error	

8.2.4 Example of Use 1

This example is supposed that there's a communication between GM7's by the user-defined protocol. The system configuration is as follows and the cable is the same with the one of 1:1 dedicated protocol communication.



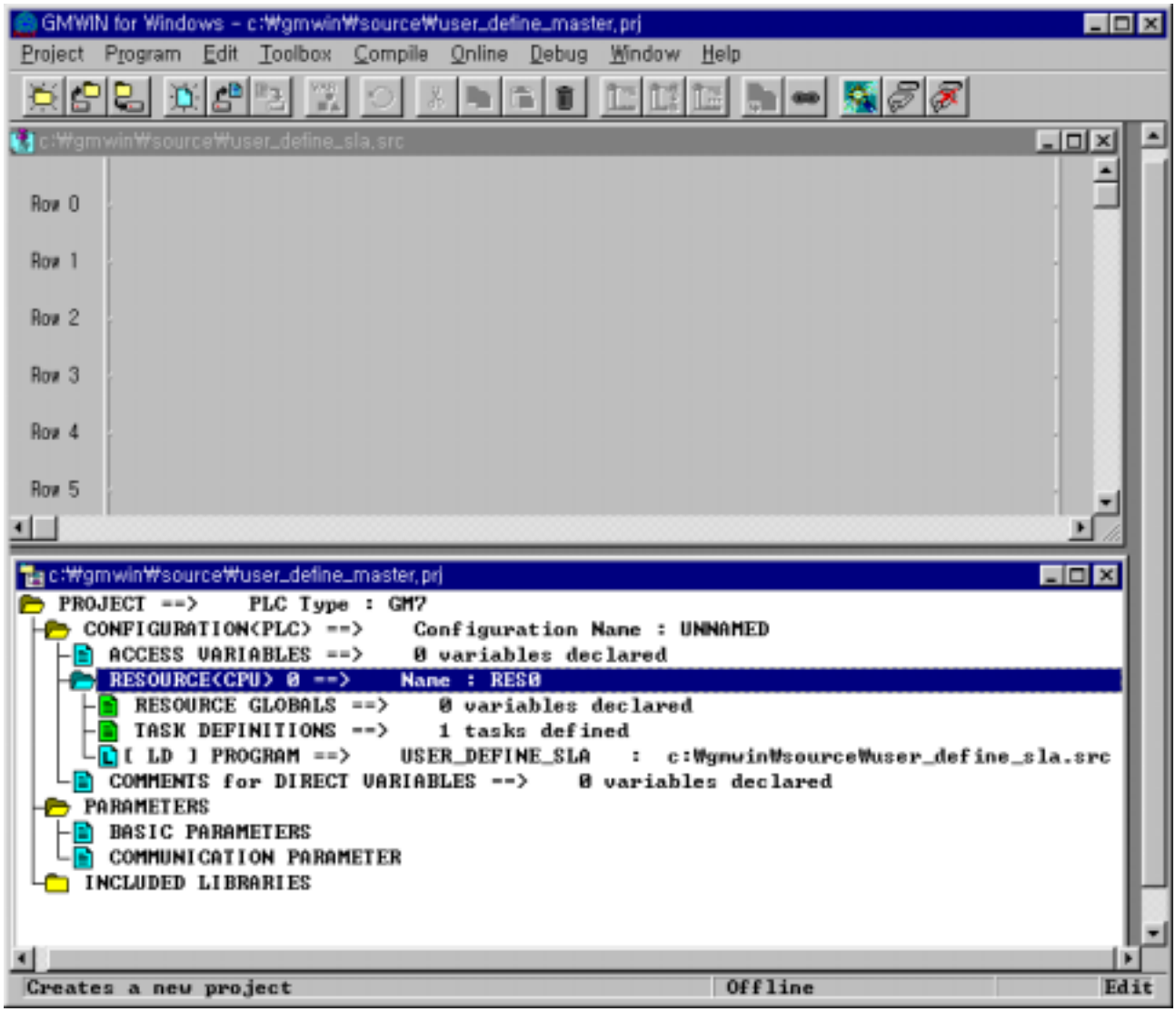
1:1 dedicated protocol communication between GM7's

The data in M area of the master station is sent to the slave station and the slave station saves the received data in M area outputs as direct variable, and sends the data back to the master. This process repeats between the master and the slave.

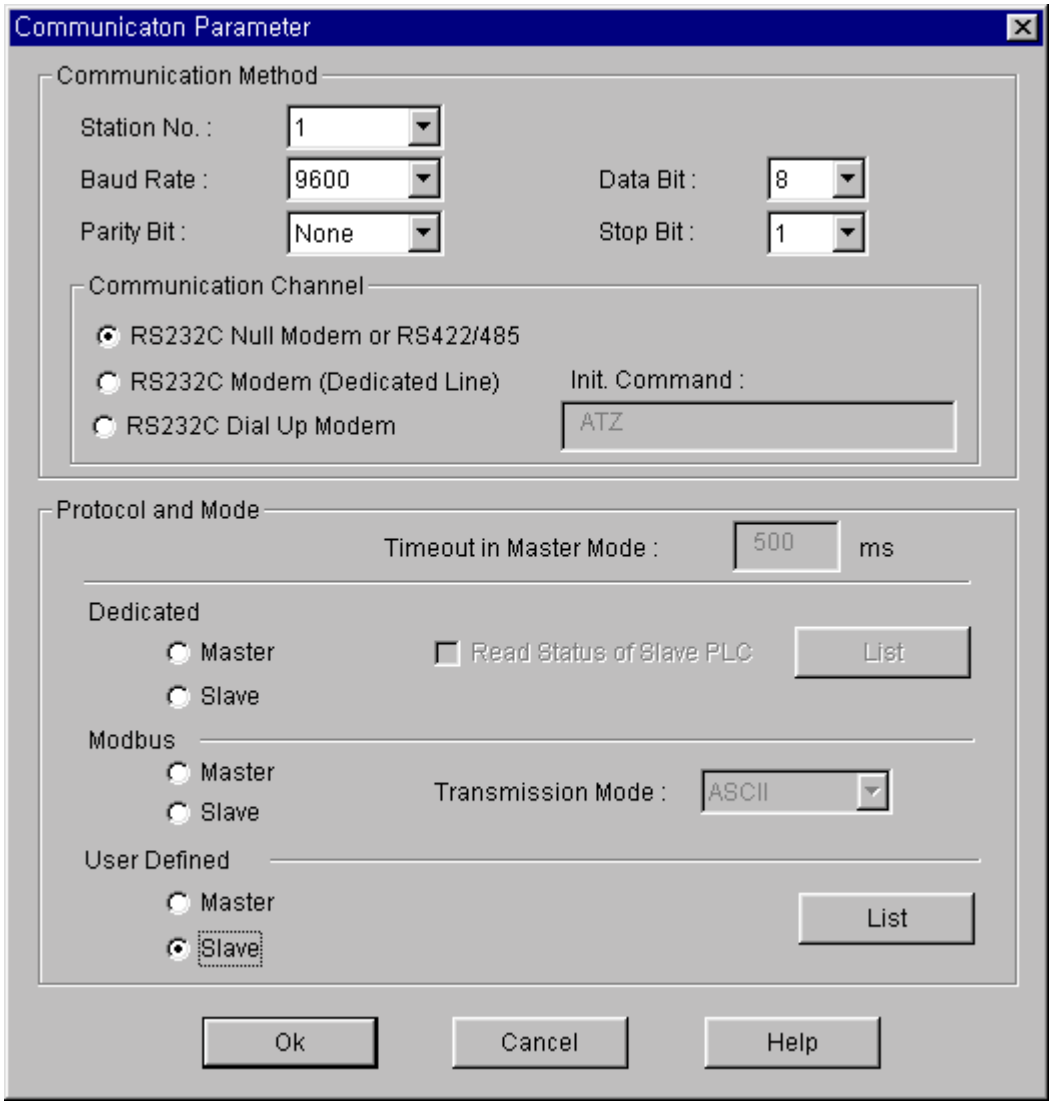
1) The Programming and setting communication parameter of the slave station

(1) Work at the slave station no. 1.

(2) Open a new project file and make a new program for the slave station.



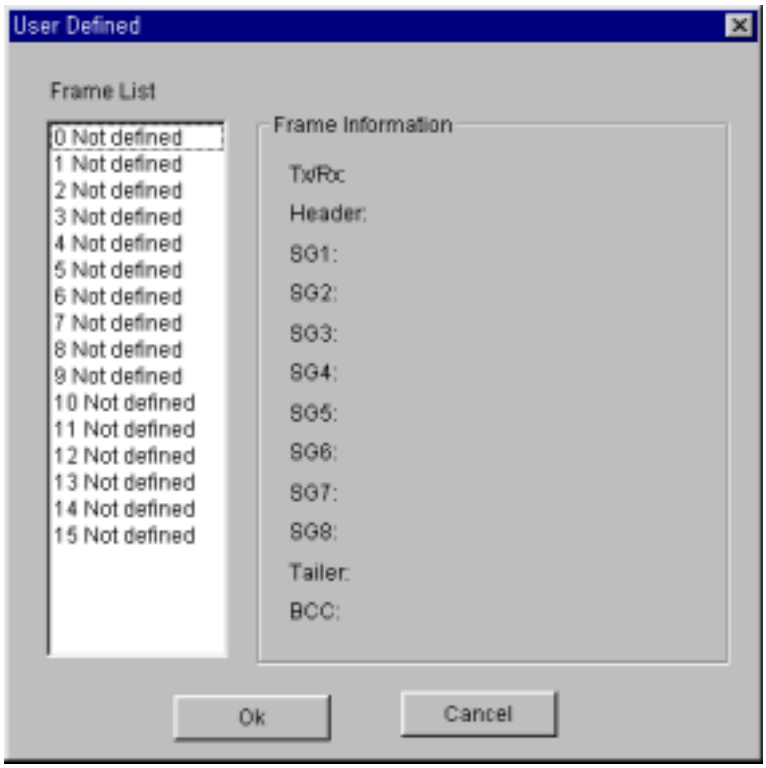
(3) Select communication parameter in GMWIN parameters and double click it, and then the window of communication parameter will be opens.



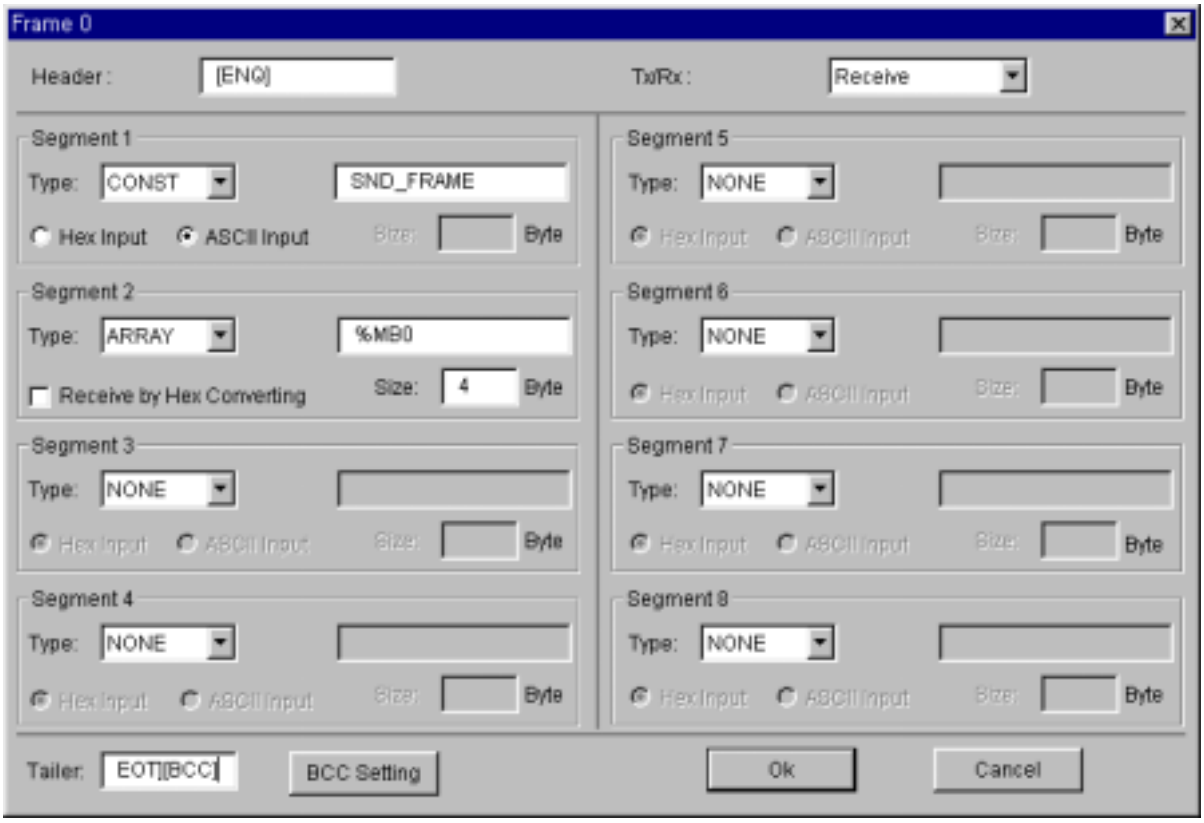
- Set parameters as the following table.

Communication Method						Protocol and Mode
Station No.	Baud Rate	Data Bit	Parity Bit	Stop Bit	Communication channel	User Defined
1	9600	8	None	1	RS232C null modem or RS422/485	Slave

(4) Click “List” to open the following window.

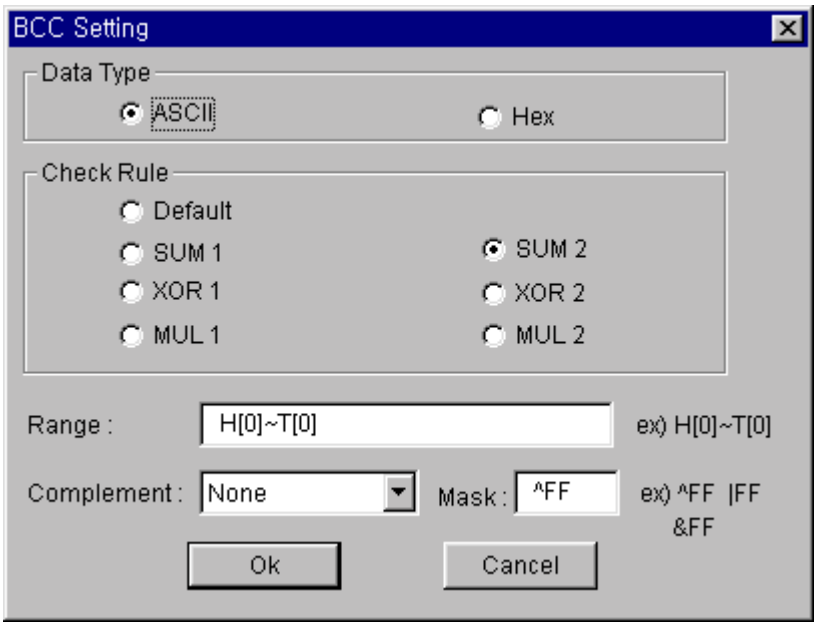


(5) Double click “Frame List 0” to activate “Frame 0” window and set as follows.



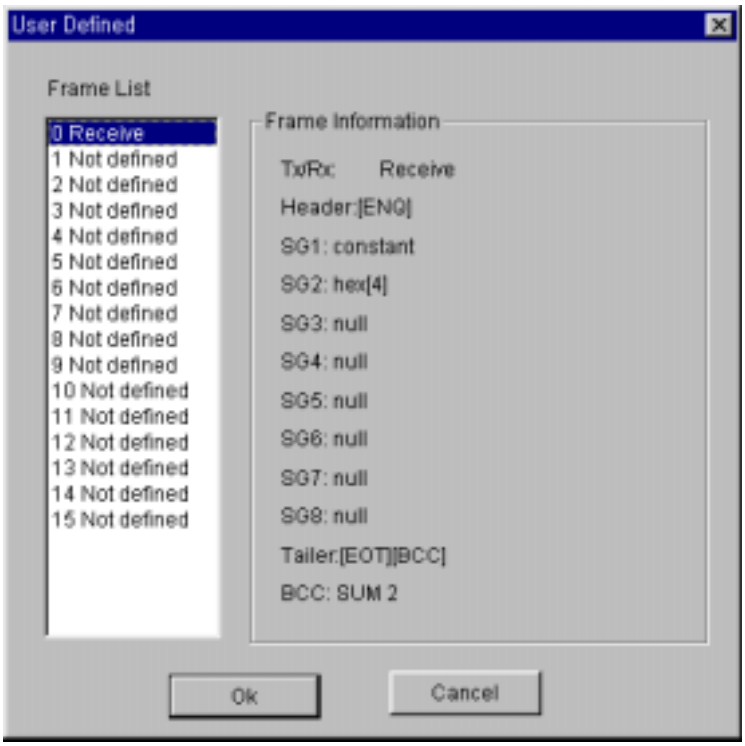
Item	Setting value
Header	[ENQ]
Tx/Rx	Receive
Segment 1	Type: CONST, field: SND_FRAME, select button for ASCII input
Segment 2	Type: ARRAY, field: %MB0, size: 4 bytes
Tailer	[EOT][BCC]

(6) "[BCC]" is set in tail after setting, click "BCC setting" to activate BCC setting window, set as follows and click "OK" to close.

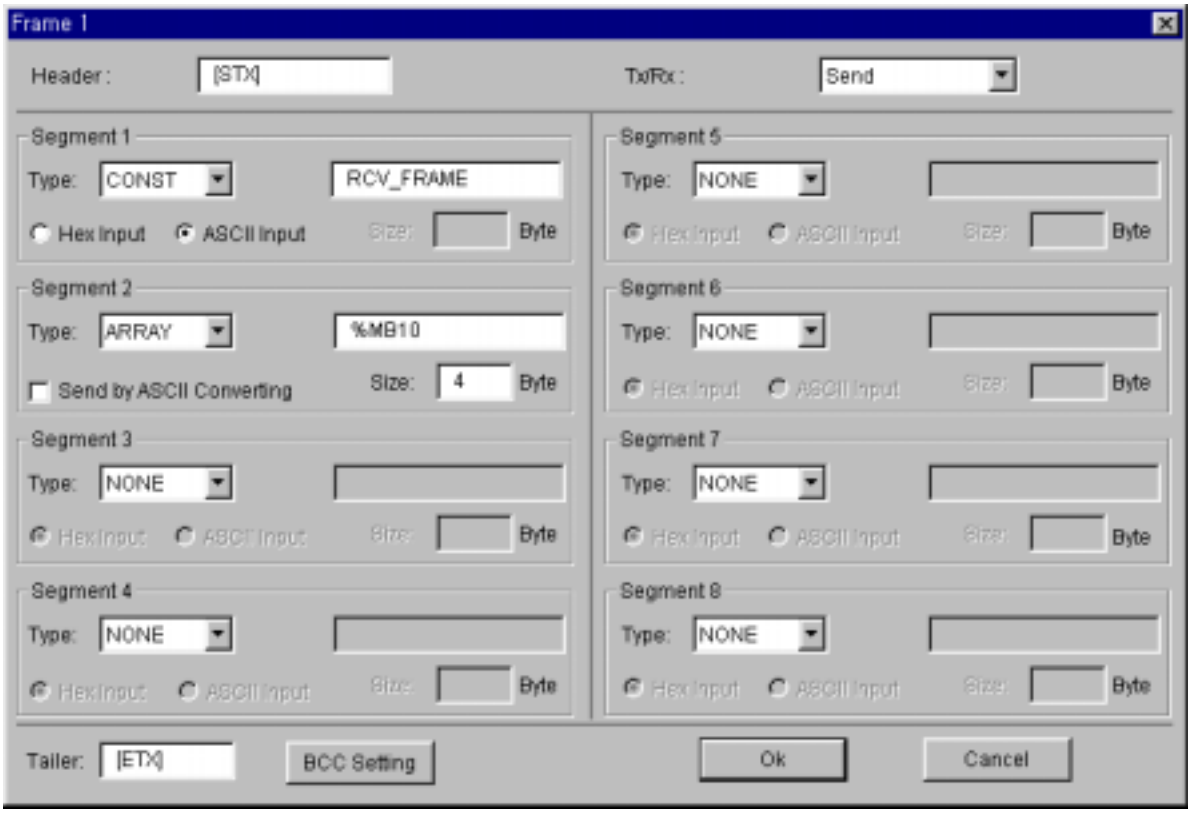


Item	Setting value
Data Type	ASCII
Check Rule	SUM 2
Range	H [0] ~ T [0] that is from header [ENQ] to tail [EOT].
Complement	None
Mask	Mask with HFF by XOR.

(7) After BCC setting, click "OK" of Frame 0 window to register the frame as follows.

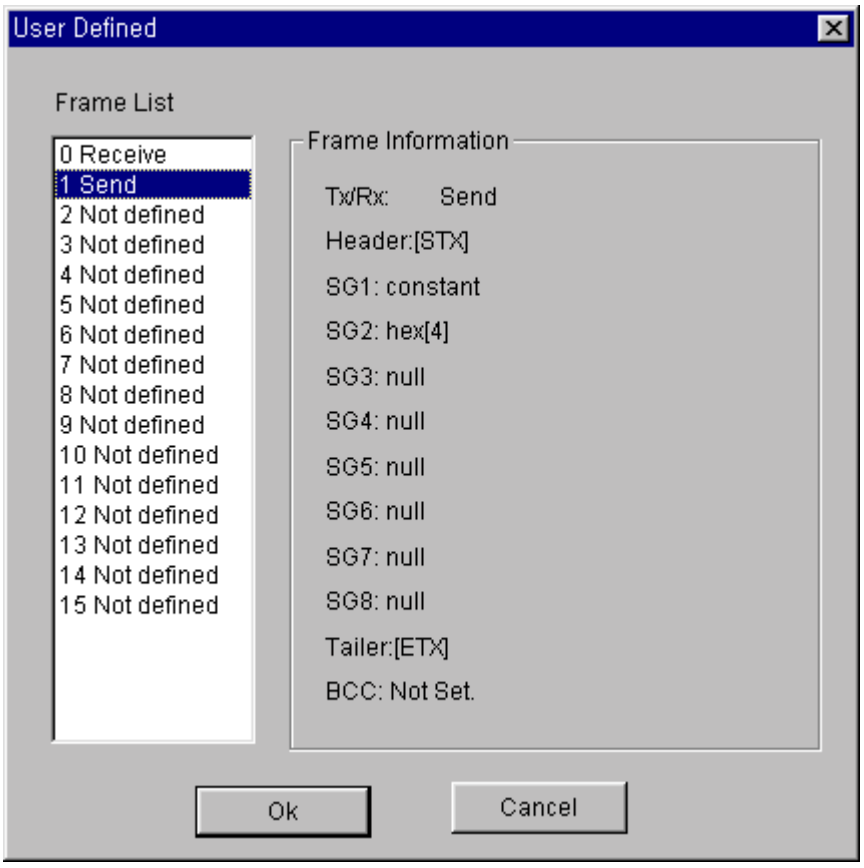


(8) Double click frame list 1 to activate Frame 1 window to set as follows.



Item	Setting value
Header	[STX]
Tx/Rx	Send
Segment 1	Type: CONST, field: RCV_FRAME, select button for ASCII input
Segment 2	Type: ARRAY, field: %MB10, size: 4 bytes
Tailer	[ETX]

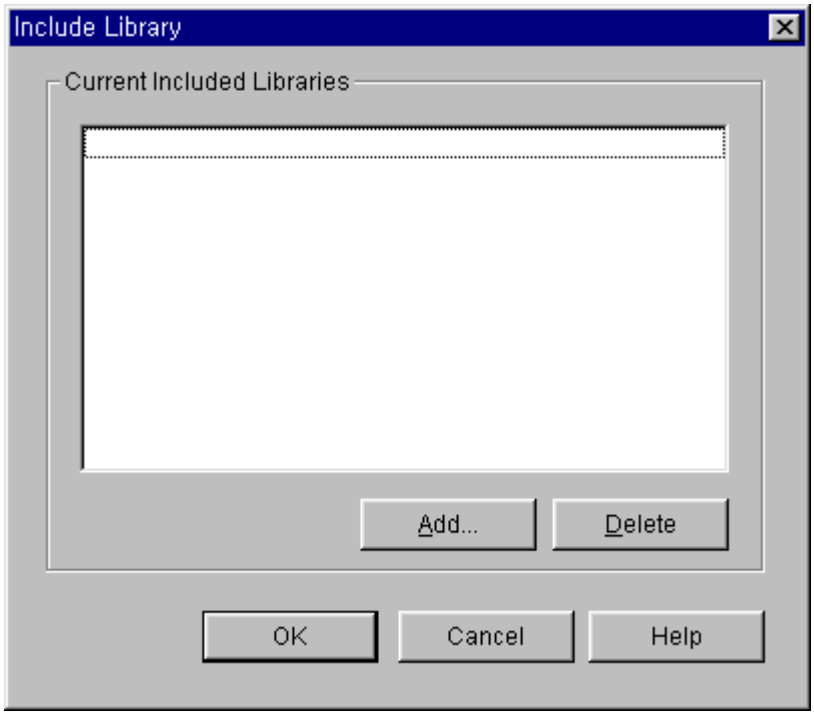
(9) After setting, click "OK" of Frame 1 to register as follows.

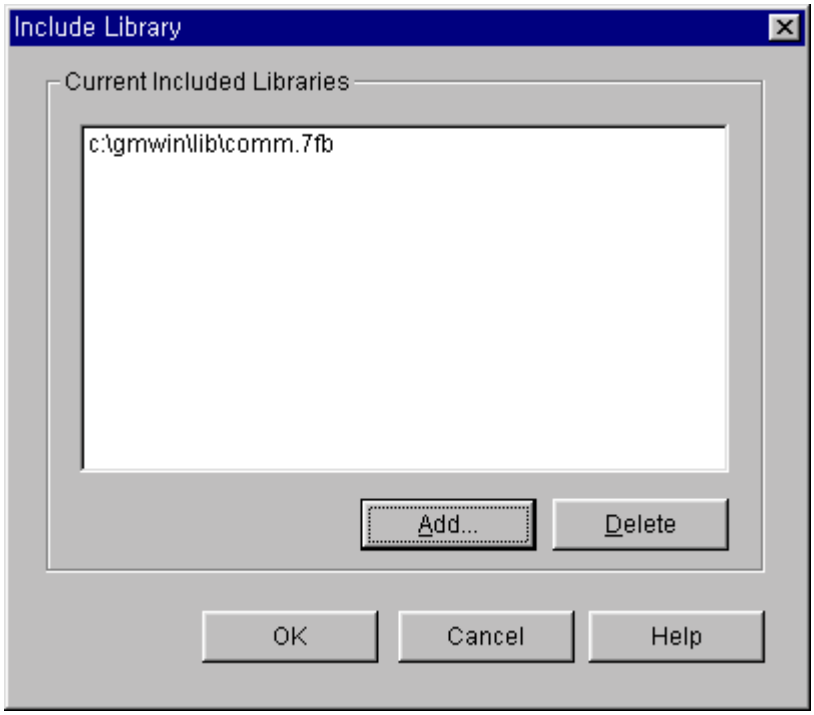


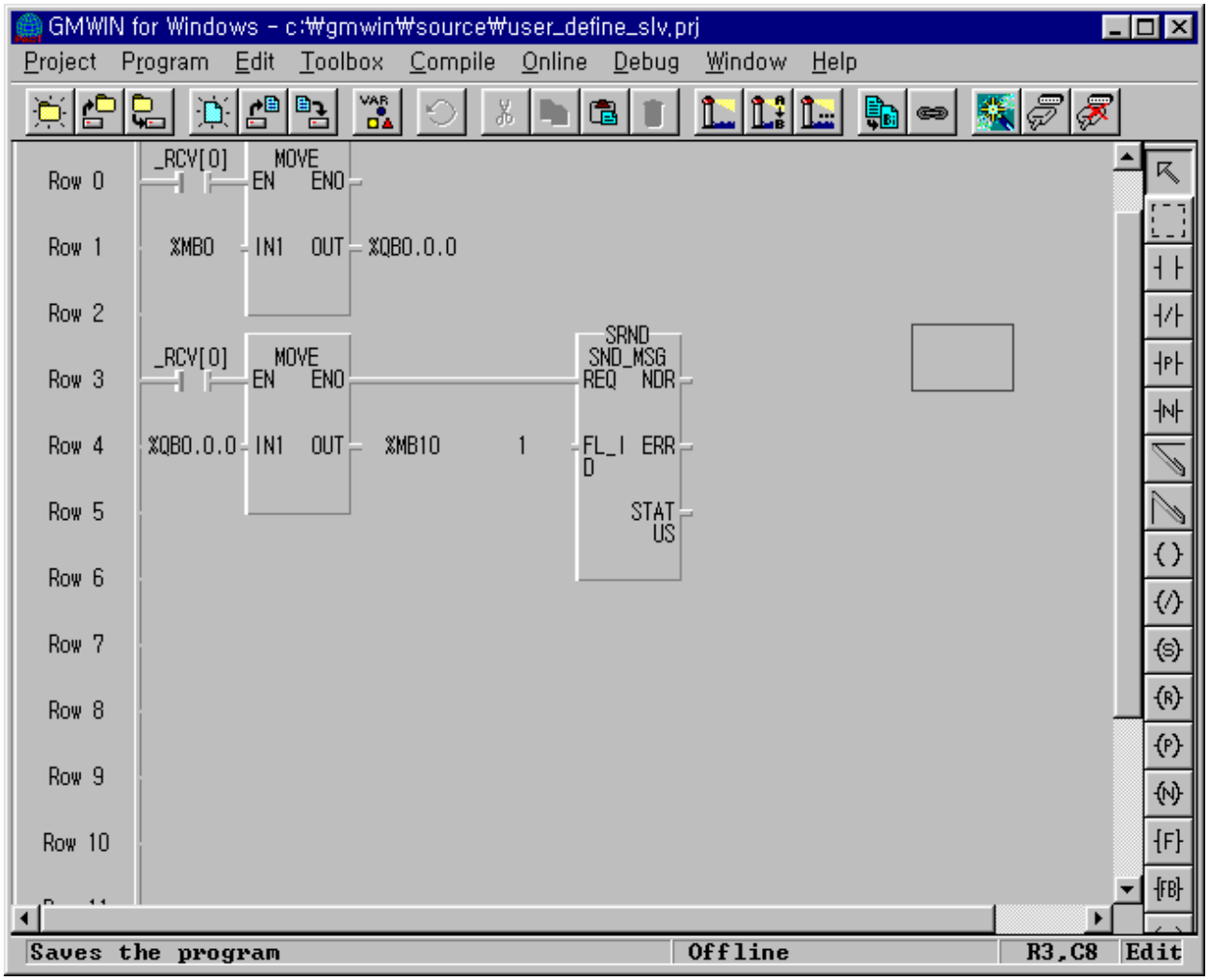
(10) Click "OK" to exit communication parameter setting and click "OK" to complete setting.

(11) Setup a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.

- Function block is used in the program. Before using function block, double click "Current included Libraries" to open the following window. Click "Add" to add COMM.7FB and click "OK."





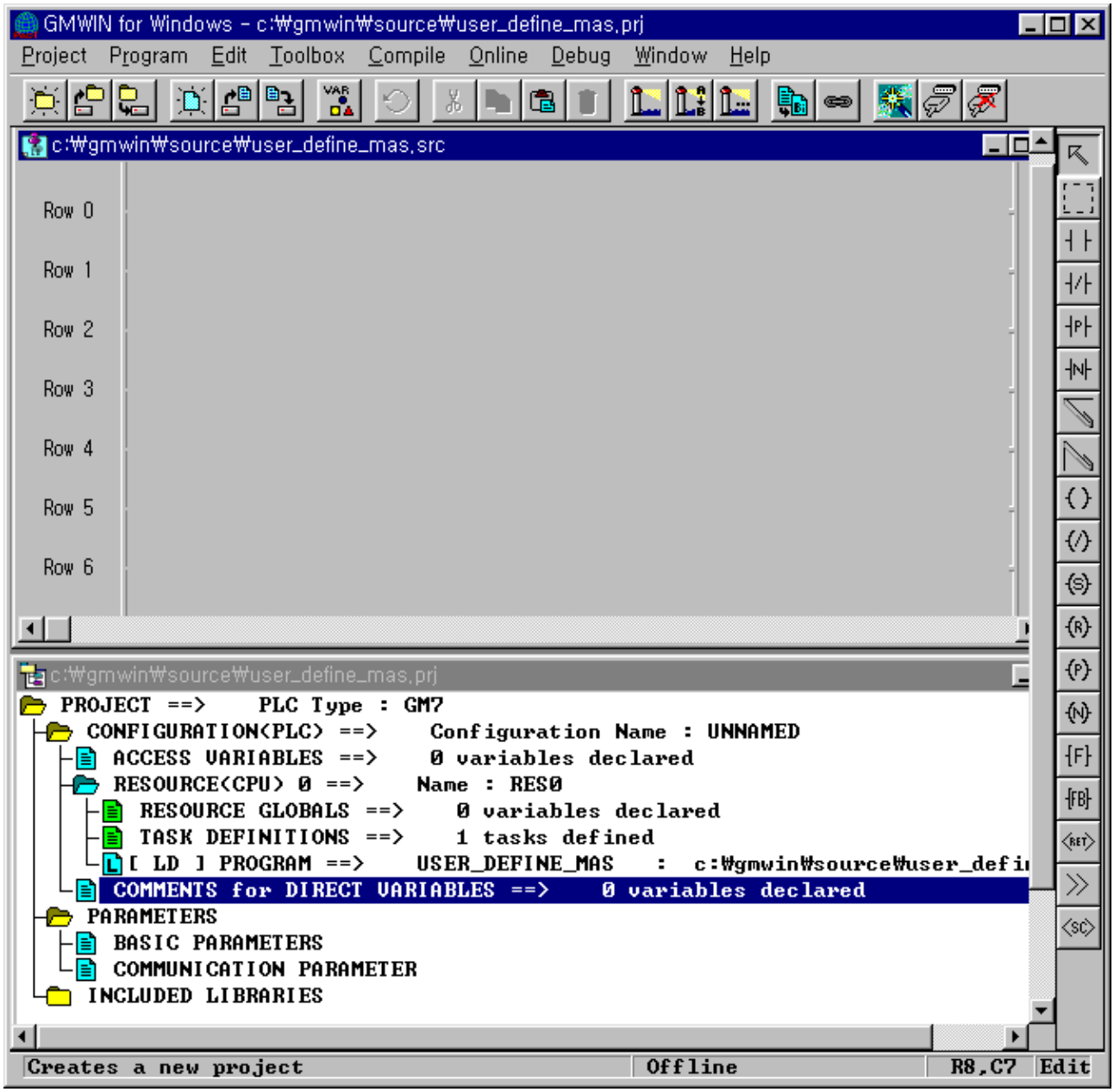


- If the frame 0 is received, save it in %MB0 of the slave (refer to frame 0 setting). Output value saved in %MB0 to direct variable %MB10 and again save the value in %QB0.0.0 in %MB10. If this execution is done without error, send frame 1 out of frame list from function block. Frame 1 is the frame that sends 4 bytes data saved in %MB10.
- If Frame 0 isn't received, this program doesn't work.
- **To open communication, set through "communication enable setting."**
- For "communication enable setting" method, refer to 3) Communication enable setting of 8.1.7 "1:1 dedicated protocol communication."

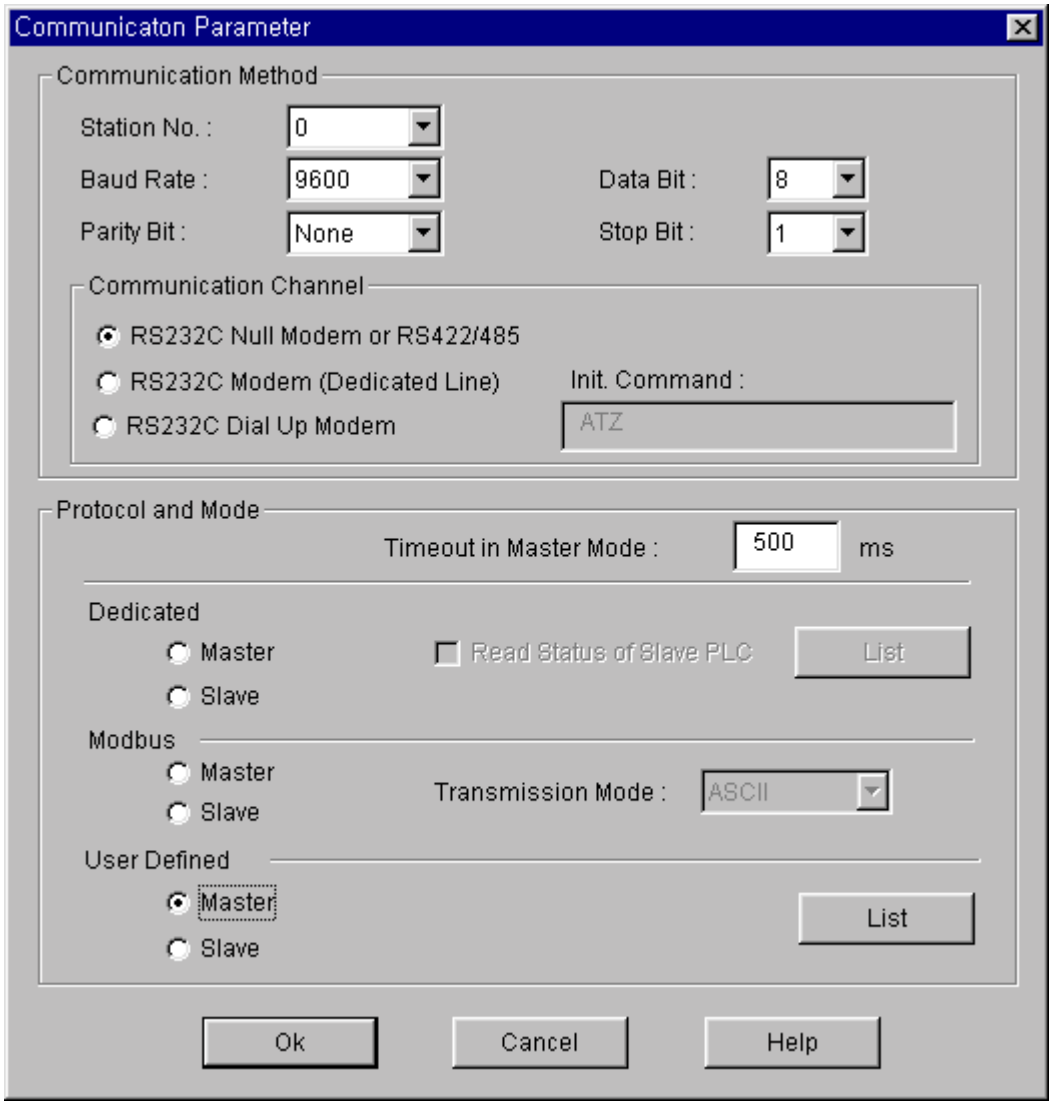
2) The programming and setting communication parameter of the master station

(1) Work at the master station no. 1.

(2) Open a new project file and set up a new program for the master station.



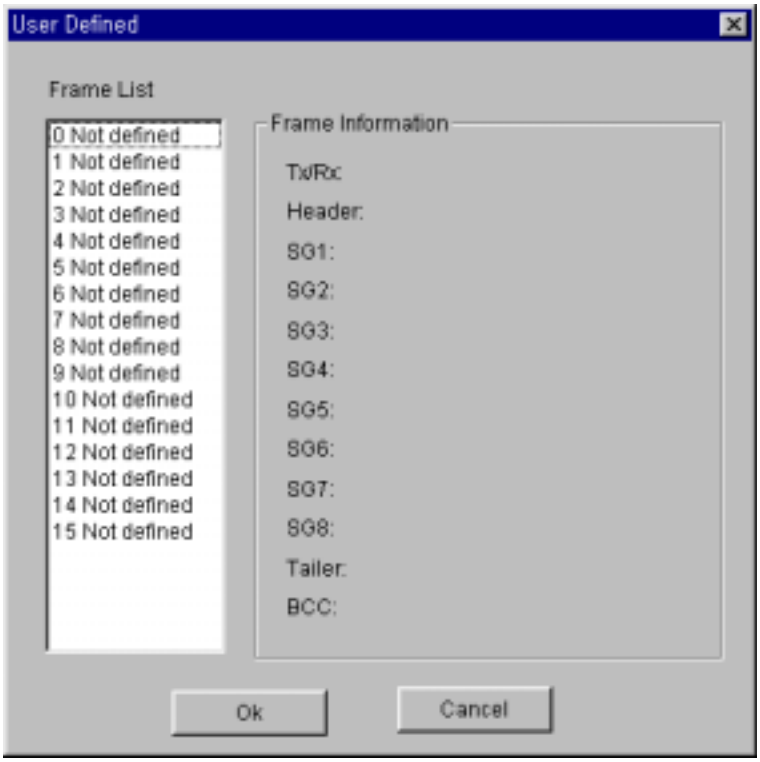
(3) If you double click the 'communication parameter' window in GMWIN, you can see the following window of the 'communication parameter.'



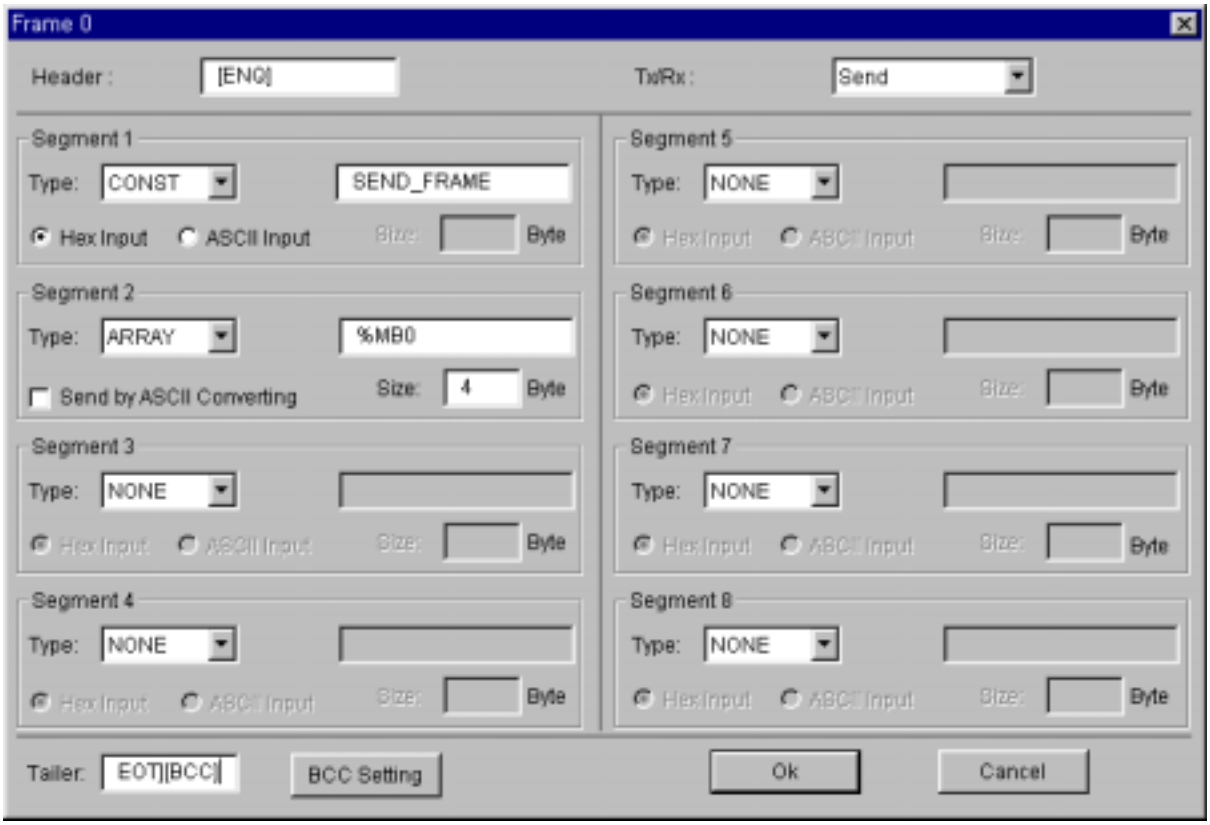
- Set parameters as the following table.

Communication Method						Protocol and Mode
Station No.	Baud Rate	Data Bit	Parity Bit	Stop Bit	Communication Channel	User Defined
0	9600	8	None	1	RS232C null modem or RS422/485	Master

(4) Click “List” to activate frame list window.

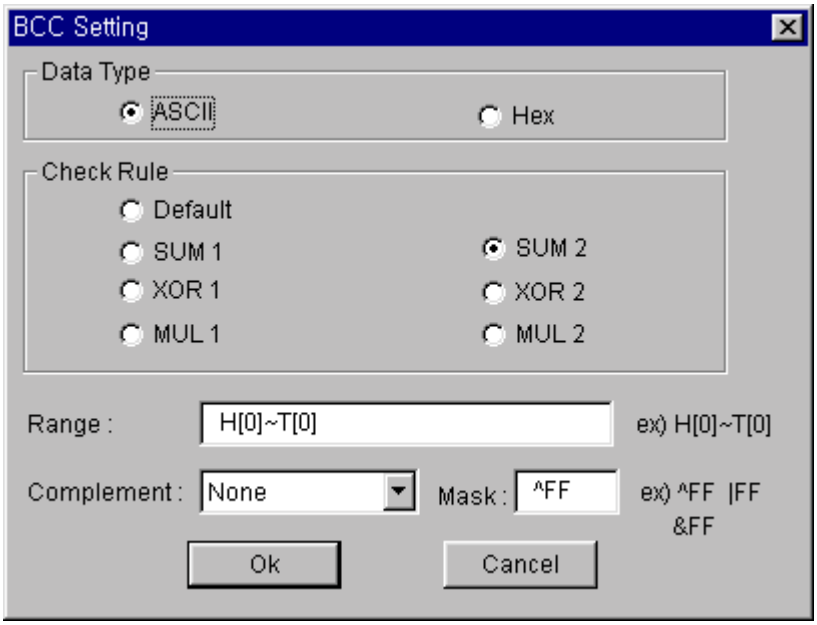


(5) Double click Frame list 0 to activate Frame 0 window and set as follows.



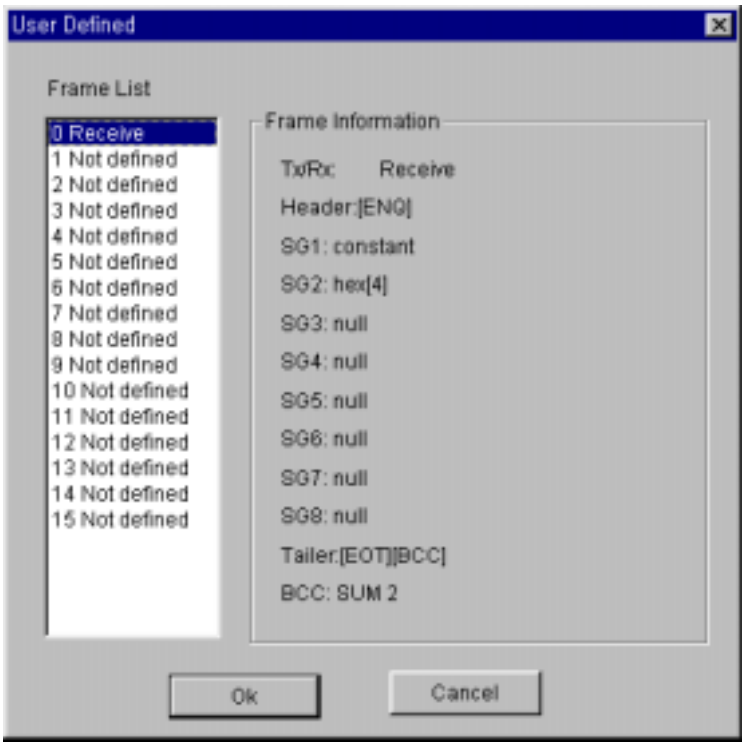
Item	Setting value
Header	[ENQ]
Tx/Rx	Send
Segment 1	Type: CONST, field: SND_FRAME, select button for ASCII input
Segment 2	Type: ARRAY, field: %MB0, size: 4 bytes
Tailer	[EOT][BCC]

(6) "[BCC]" is set in tail after setting, click "BCC Setting" to activate BCC setting window, set as follows and click "OK" to close.

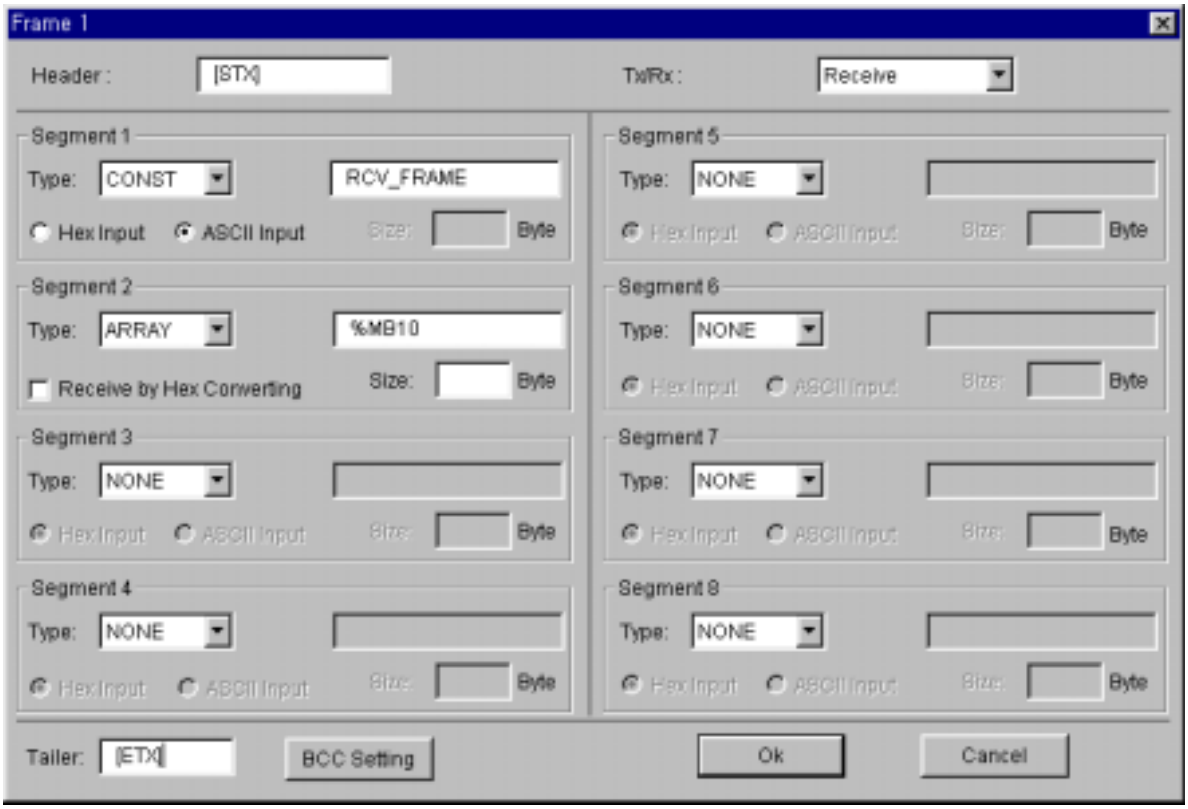


Item	Setting value
Data Type	ASCII
Check Rule	SUM 2
Range	H [0] ~ T [0] that is from header [ENQ] to tail [EOT].
Complement	None
Mask	Mask with HFF by XOR.

(7) After BCC setting, click "OK" of Frame 0 window to register the frame as follows.

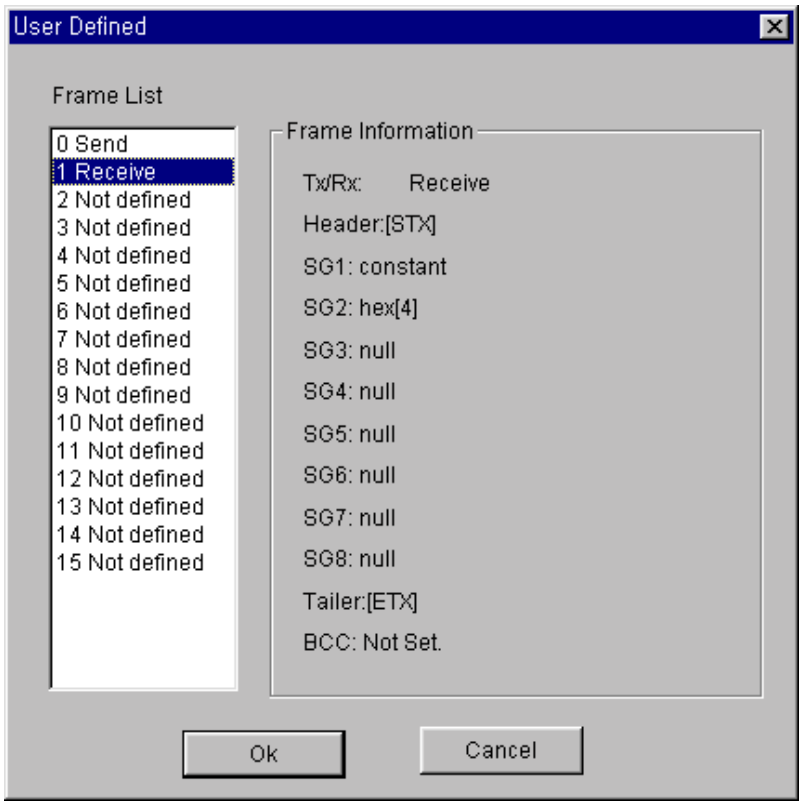


(8) Double click frame list 1 to activate Frame 1 window to set as follows.



Item	Setting value
Header	[STX]
Tx/Rx	Receive
Segment 1	Type: CONST, field: RCV_FRAME, select button for ASCII input
Segment 2	Type: ARRAY, field: %MB10, size: 4 bytes
Tailer	[ETX]

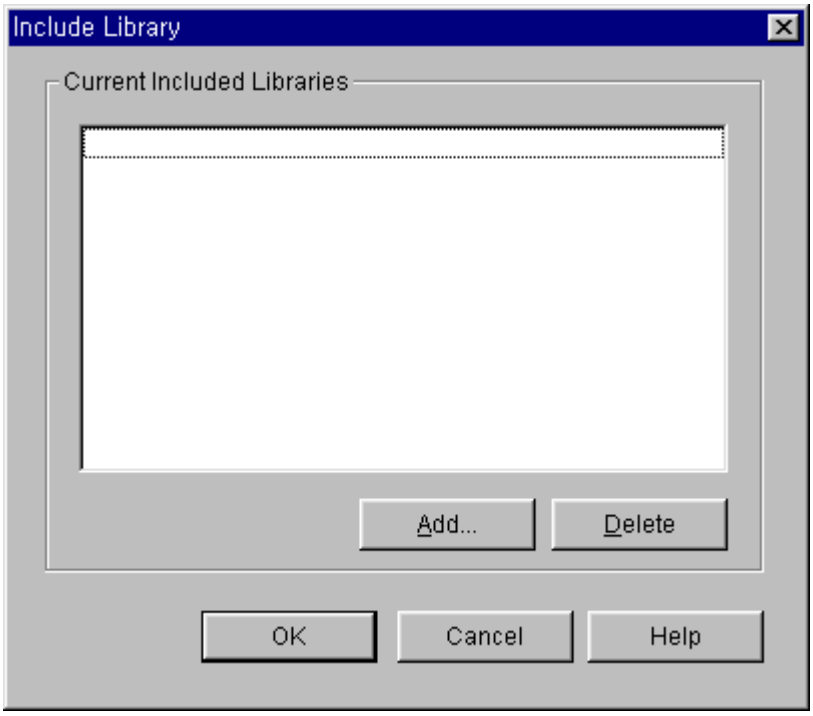
(9) After setting, click “OK” of Frame 1 to register as follows.

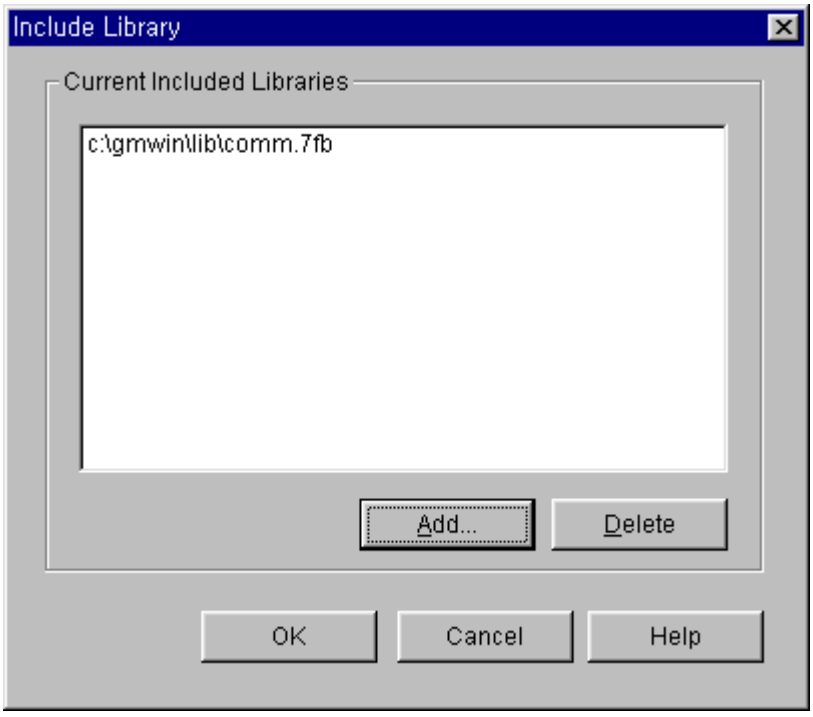


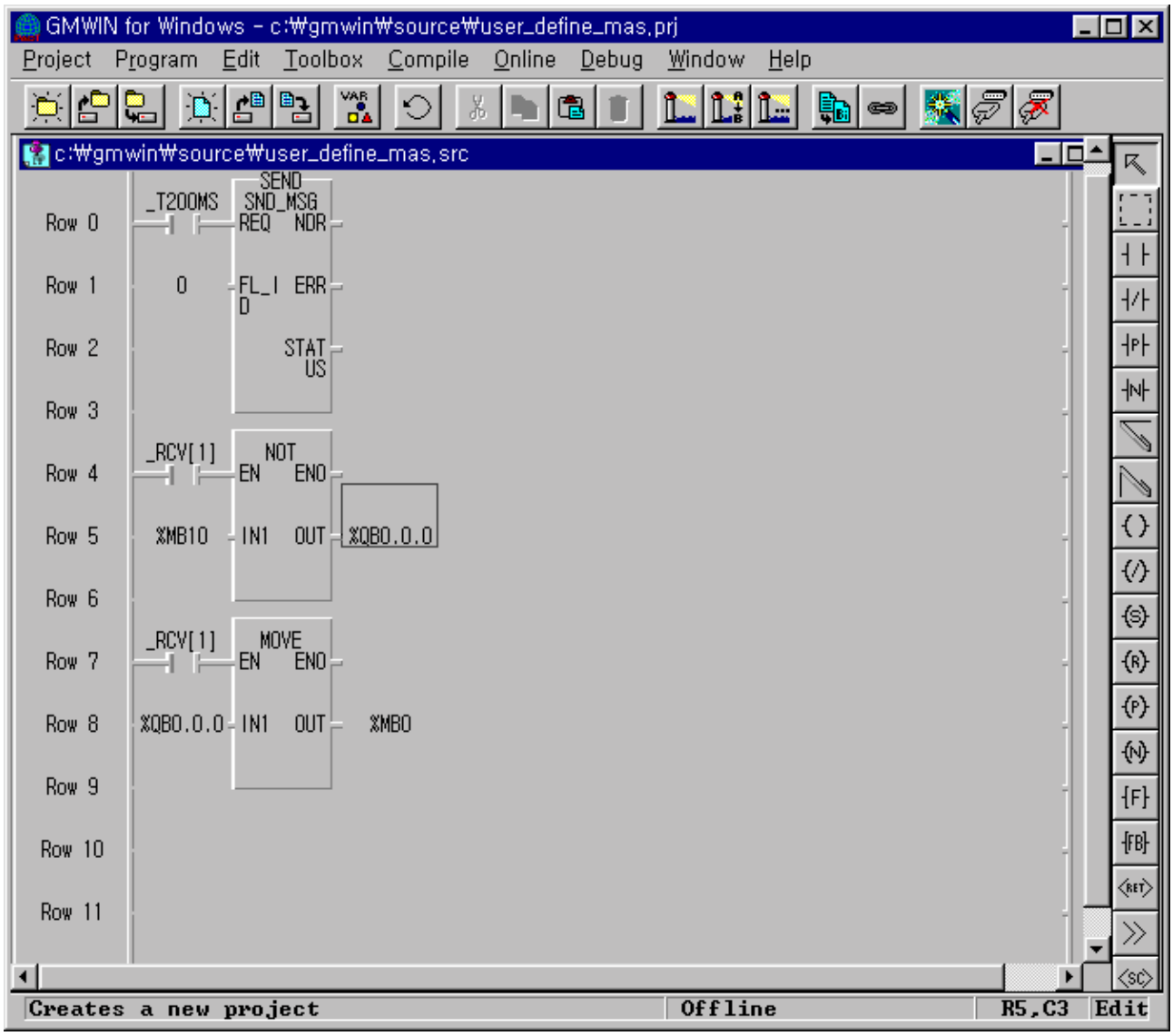
(10) Click “OK” to exit communication parameter setting and click “OK” to complete setting.

(12) Setup a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.

- Function block is used in the program. Before using function block, double click “Current included Libraries” to open the following window. Click “Add” to add COMM.7FB and click “OK.”



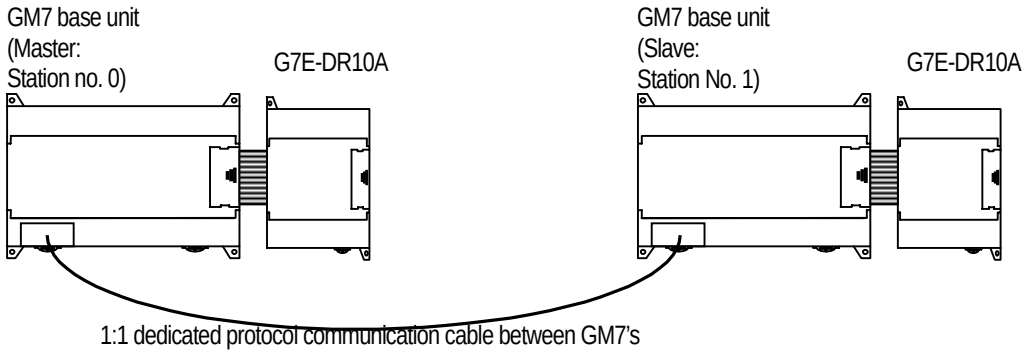




- It sends Frame 0 by operating function block every 200ms.
- Frame 0 sends 4-byte value from the master station in %MB0 to the slave station.
- The slave station saves Frame 1 that the master sent to %MB10. (Refer to frame setting of Frame 1) It logically reverses the value of %MB10 that is saved at this moment, outputs to the direct variable, %QB0.0.0 and saves again the value output to the direct variable %QB0.0.0. to %MB0.
- Conclusively, when 8 LED of the master are on, then 8 LED of the slave are off, and vice versa.
- **To open communication, set through "communication enable setting."**
- For "communication enable setting" method, refer to 3) Communication enable setting of 8.1.7 "1:1 dedicated protocol communication."

8.2.5 Example of Use 2

This example is supposed to activate direct variables with the help of the user-defined protocol. It's ideal for the system configuration to connect digital I/O module to GM7 base unit. If digital I/O module isn't connected, only a part of the operation is possibly checked.



1) The Program and communication parameter of the slave station

(1) Work at the slave station no. 1.

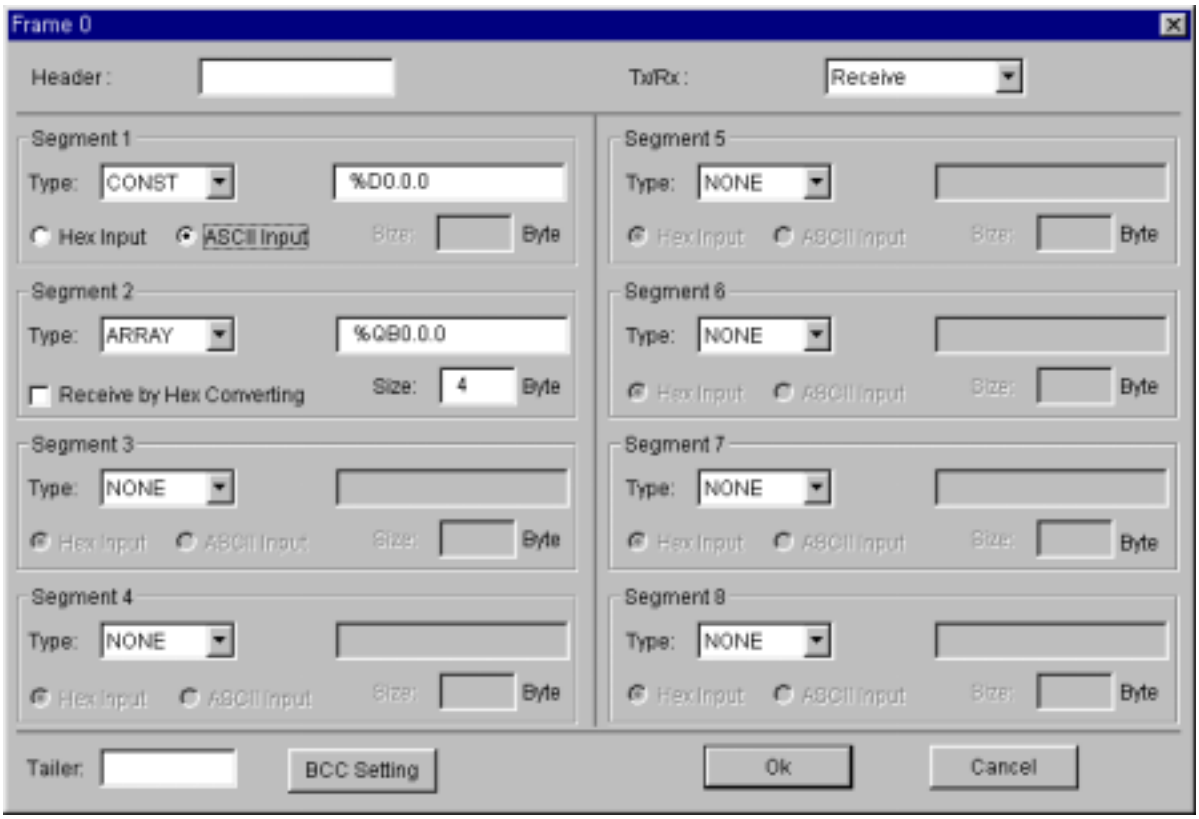
(2) Open a new project file and make a new program for the slave station.

(3) Select communication parameter in GMWIN parameter and double-click it, the window of comm. parameter menu opens.

- Set parameters as the following table.

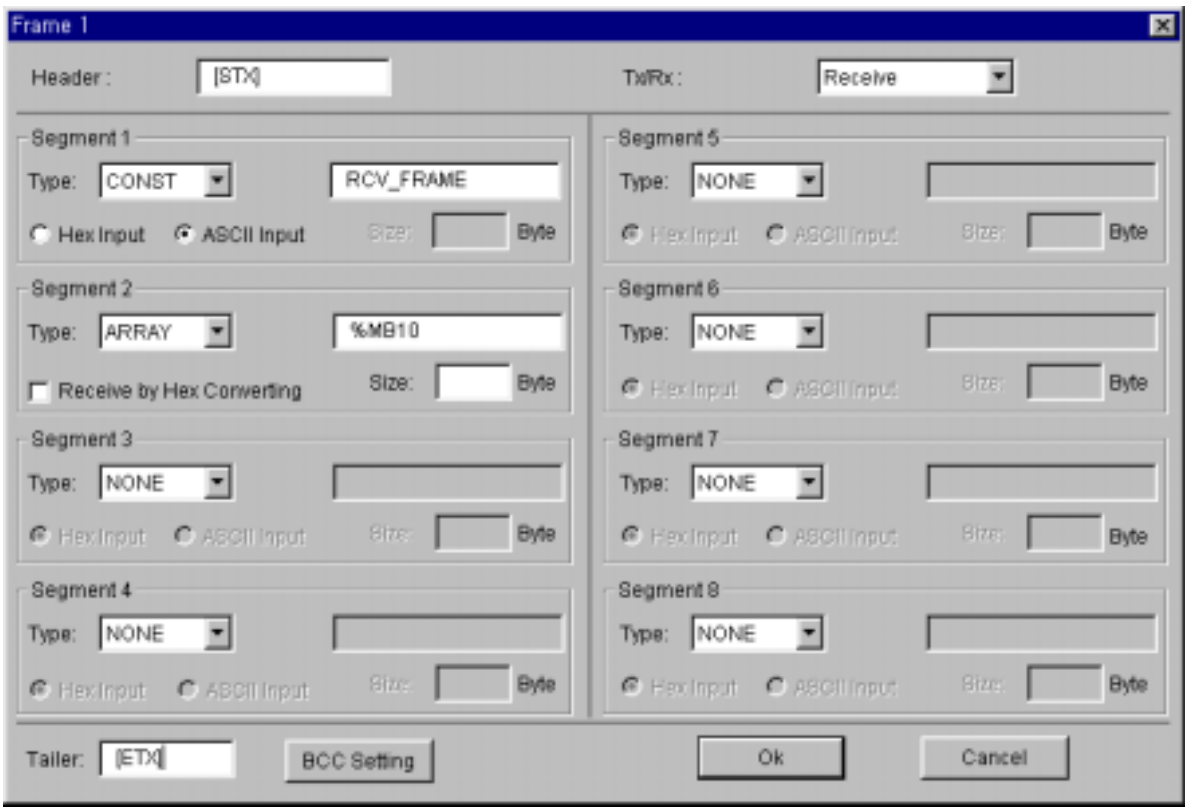
Communication Method						Protocol and Send mode
Station no.	Baud rate	Data bit	Parity bit	Stop bit	Communication channel	Dedicated
1	9600	8	None	1	RS232C null modem or RS422/485	Slave

(4) Double “frame list 0” to activate “Frame 0” window and set as follows.



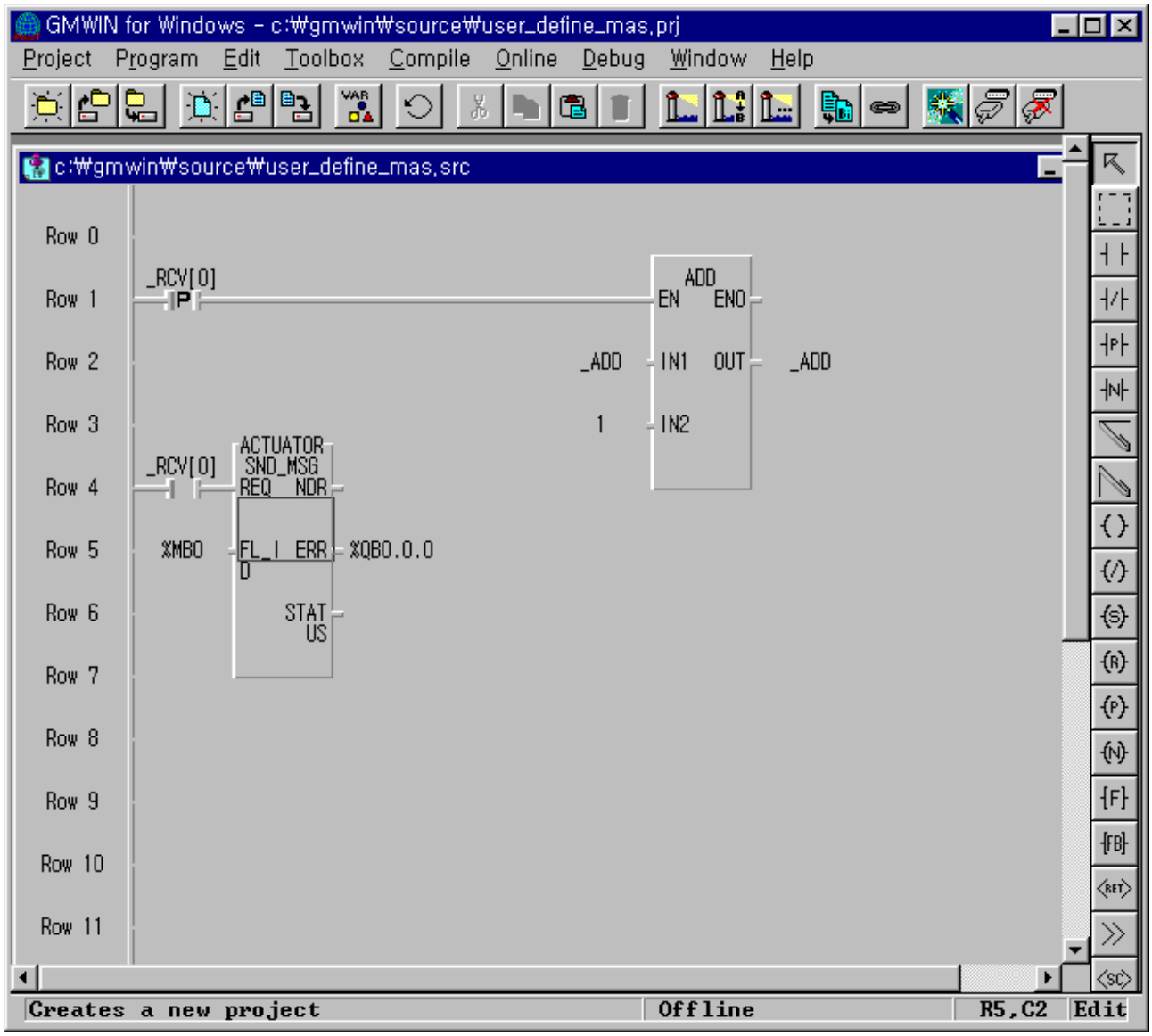
Item	Setting value
Header	None
Send/receive	Receive
Segment 1	Type: CONST, field: %QD0.0.0, select button for ASCII input
Segment 2	Type: ARRAY, field: %QB0.0.0, size: 4 bytes
Tail	None

(5) Double click frame list 1 to activate Frame 1 window to set as follows.



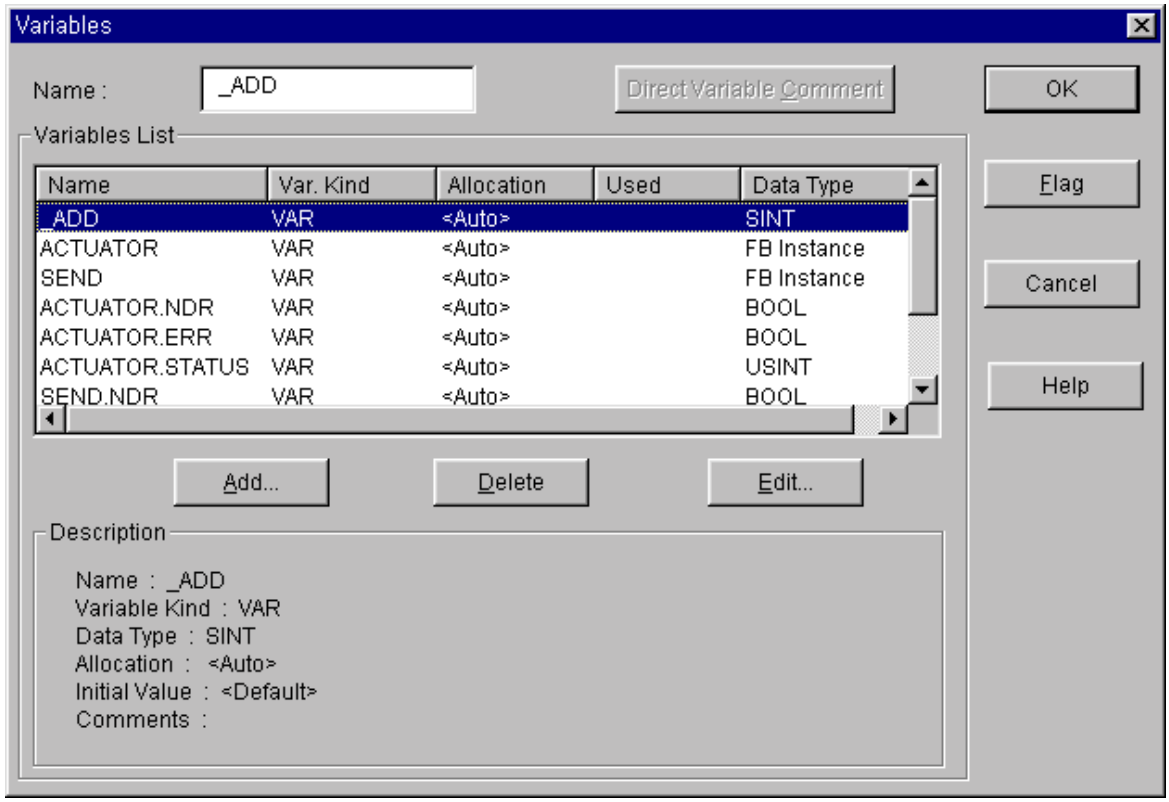
Item	Setting value
Header	None
Send/receive	Send
Segment 1	Type: CONST, field: %QB0.1.0, select button for ASCII input
Segment 2	Type: ARRAY, field: %QB0.1.0, size: 1 bytes
Tail	None

(6) Set up a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.



- When the frame 0 is received, it outputs the value to %QB0.0.0 of the slave station. (Refer to the Frame 0 setting.)
- When Frame 0 is received, it operates ADD function to increase the value of the symbolic variable, _ADD 1 by 1, and because the symbolic variable _ADD has been assigned a memory at the direct variable, %QB0.1.0, outputs the result of the ADD function to the digital I/O module.
- The slave reads the direct variable, %QB0.1.0 of the digital I/O and sends it again to the master. (This function enables to check whether the communication is being done without the digital I/O module by the direct variable outputs of the GM7 base unit.)
- **To open communication, set through "communication enable setting."**

- Variables window



2) The Program and communication parameter of the master station

(1) Work at the master station no. 0.

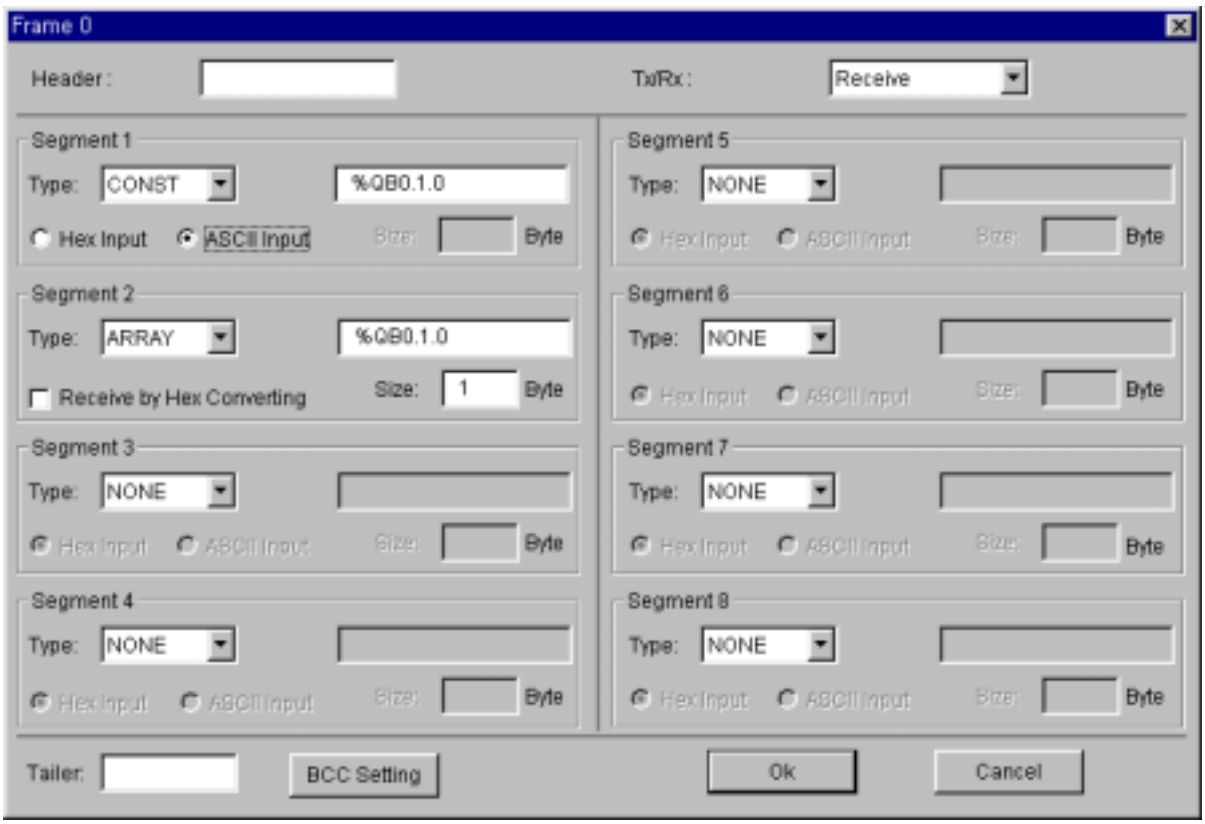
(2) Open a new project file and make a new program for the master station.

(3) Select communication parameter in GMWIN parameter and double-click it, the window of comm. parameter menu opens.

- Set parameters as the following table.

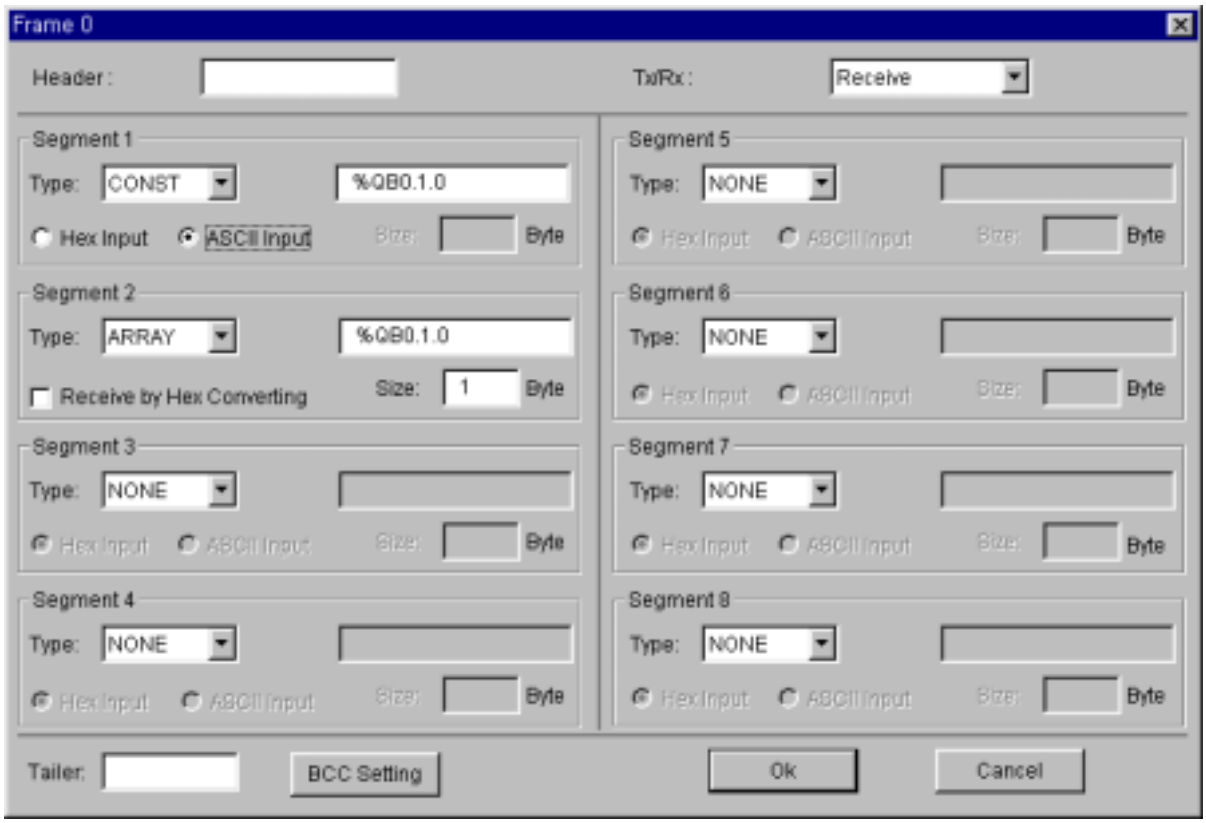
Communication Method						Protocol and Send mode
Station no.	Baud rate	Data bit	Parity bit	Stop bit	Communication channel	User-defined
0	9600	8	None	1	RS232C null modem or RS422/485	Master

(4) Double “frame list 0” to activate “Frame 0” window and set as follows.



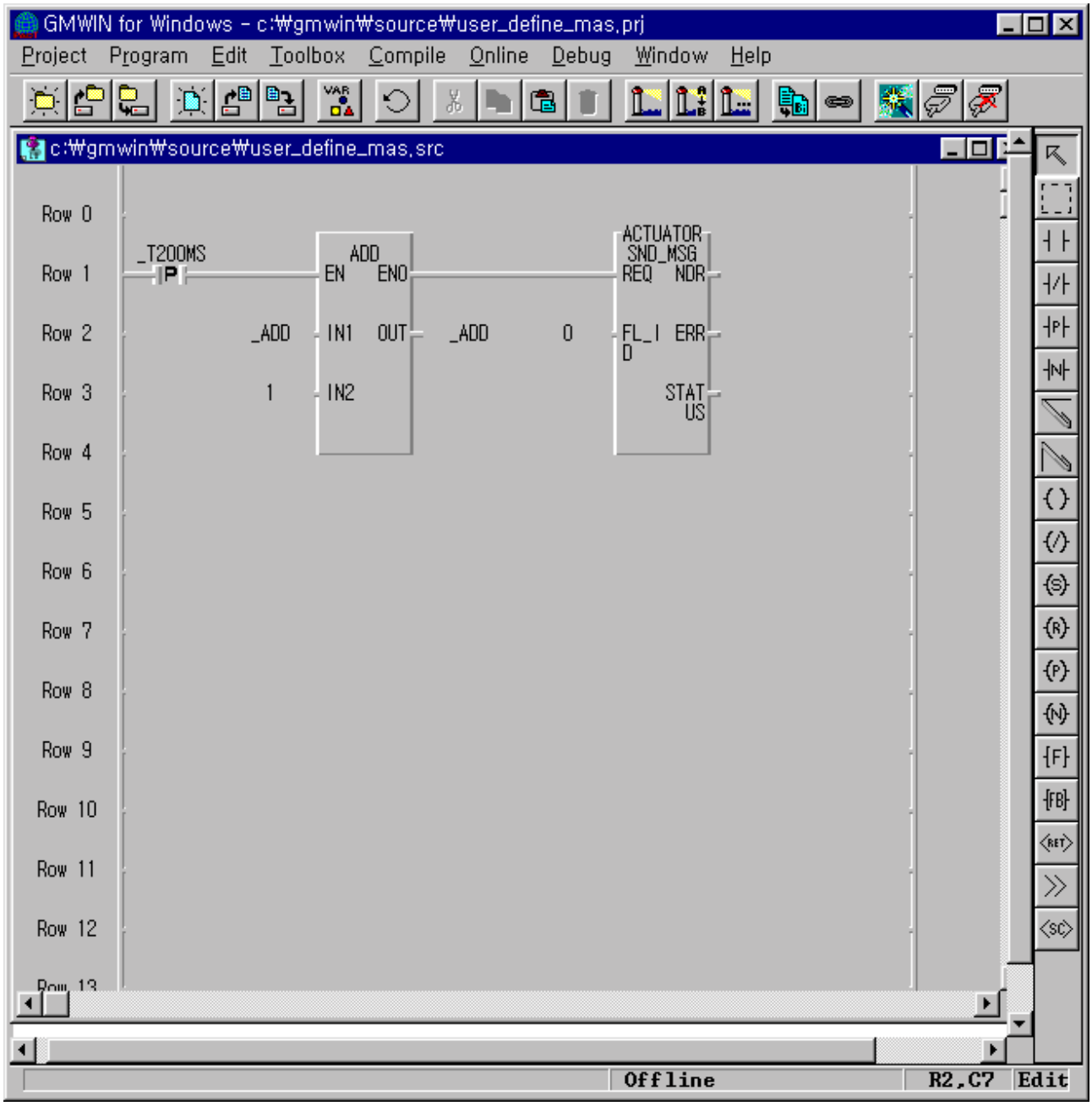
Item	Setting value
Header	None
Send/receive	Send
Segment 1	Type: CONST, field: %QB0.1.0, select button for ASCII input
Segment 2	Type: ARRAY, field: %QB0.1.0, size: 1 bytes
Tail	None

(5) Double click frame list 1 to activate Frame 1 window to set as follows.



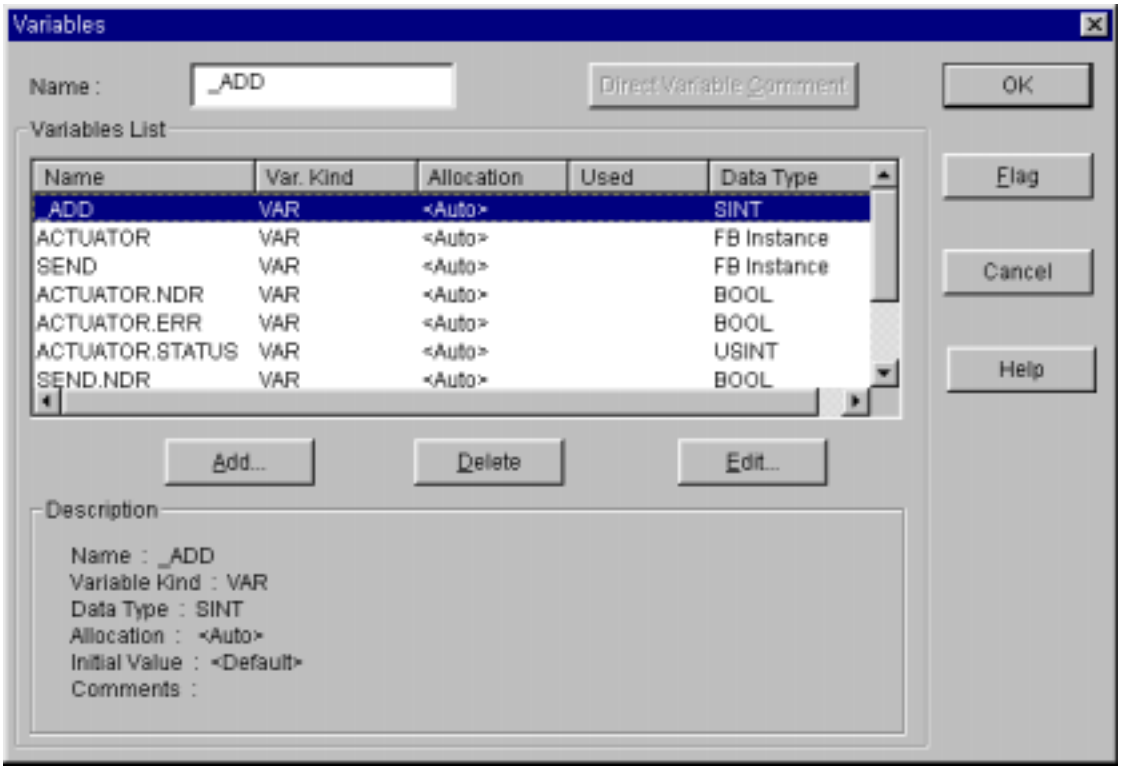
Item	Setting value
Header	None
Send/receive	Receive
Segment 1	Type: CONST, field: %QB0.1.0, select button for ASCII input
Segment 2	Type: ARRAY, field: %QB0.1.0, size: 1 bytes
Tail	None

(6) Set up a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.



- It operates ADD function to increase the value of the symbolic variable, _ADD 1 by 1 at every 20ms, and because the symbolic variable _ADD has been assigned a memory at the direct variable, %QB0.0.0, outputs the result of the output contact point of the GM7 base unit.
- The master reads the direct variable, %QB0.0.0 of the GM7 base unit and sends it again to the master. (Refer to Frame 0 setting.)
- **To open communication, set through "communication enable setting."**

- Variables window



8.3 Modbus Protocol Communication

8.3.1 Introduction

GM7 built-in communication supports Modbus, the Modicon product's communication protocol. It supports ASCII mode, using ASCII data and RTU mode using Hex data. Function code used in Modbus is supported by function block and especially function code 01, 02, 03, 04, 05, 06, 15, and 16. Refer to "Modicon Modbus Protocol Reference Guide"(http://www.modicon.com/techpubs/toc7.html).

8.3.2 Basic Specification

1) ASCII mode
   (1) It communicates, using ASCII data.
   (2) Each frame uses ': (colon : H3A)', for header, CRLF (Carriage Return-Line Feed : H0D H0A), for tail.
   (3) It allows Max. 1 second interval between characters.
   (4) It checks errors, using LRC.
   (5) Frame structure (ASCII data)

Item	Header	Address	Function code	Data	LRC	Tail (CR LF)
Size	1 byte	2 bytes	2 bytes	n bytes	2 bytes	2 bytes

2) RTU mode
   (1) It communicates, using hex data.
   (2) There's no header and tail. It starts with address and finishes frame with CRC.
   (3) It has at least 3.5 character times between two frames.
   (4) It ignores the current frame when 1.5 character times elapse between characters.
   (5) It checks errors, using 16 bit CRC.
   (6) Frame structure (hex data).

Item	Address	Function code	Data	CRC
Size	1 byte	1 bytes	n bytes	2 bytes

REMARK

1) The size constituting 1 letter is 1 character. So 1 character is 8 bits that is 1 byte.

2) 1 character time means the time lapsed for sending 1 character.

   Ex) Calculation of 1 character time at 1200 bps.

   1200 bps means that it takes 1 sec to send 1200 bits. To send 1 bit, 1 sec/1200 bits = 0.83 ms. Therefore 1 character time is 0.83ms * 8 bits = 6.64ms.

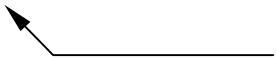
3) 584, 984 A/B/X executes frame division, using intervals of more than 1 sec without LRC in processing internally.

3) Address area
   (1) Setting range is available from 1 to 247, but GM7 supports from 0 to 31.
   (2) Address 0 is used for broadcast address. Broadcast address is all slave device recognize and respond to like the self-address, which can't be supported by GM7.

4) Function code area
   (1) GM7 supports only 01, 02, 03, 04, 05, 06, 15, and 16 among Modicon products' function codes.
   (2) If the response format is confirm+(ACK), it uses the same function code.
   (3) If the response format is confirm-(NCK), it returns as it sets the 8th bit of function code as 1.

   Ex) If function code is 03, (we write here only function code part. Because only function codes are different.)

[Request]	0000 0011 (H03)
[Confirm+]	0000 0011 (H03)
[Confirm-]	1000 0011 (H83)



It returns as it sets the 8th bit of function code of request frame.

5) Data area
   (1) It sends data, using ASCII data(ASCII mode) or hex (RTU mode).
   (2) Data is changed according to each function code.
   (3) Response frame uses data area as response data or error code.

6) LRC Check/CRC Check area
(1) LRC (Longitudinal Redundancy Check) : It works in ASCII mode. It takes 2' complement from sum of frame except header or tail to change into ASCII code,
(2) CRC (Cyclical Redundancy Check): It works in RTU mode. It uses 2-byte CRC check rules.

REMARK

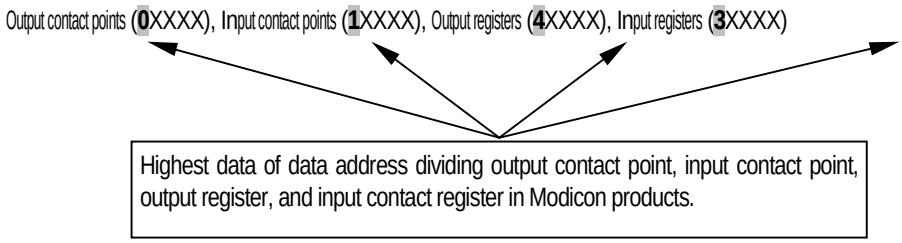
1) All numerical data can use hexadecimal, decimal, and binary type. If we convert decimal 7 and 10 into each type:
   Hexadecimal : H07, H0A or 16#07, 16#0A
   Decimal : 7, 10
   Binary : 2#0111, 2#1010

7) Function code types and memory mapping

Code	Function code name	Modicon PLC Data address	G SERIES-mapping	Remark
01	Read Coil Status	0XXXX(bit-output)	%MX0~%MX9999	Read bits
02	Read Input Status	1XXXX(bit-input)	%MX0~%MX9999	Read bits
03	Read Holding Registers	4XXXX(word-output)	%MW0~%MW9999	Read words
04	Read Input Registers	3XXXX(word-input)	%MW0~%MW9999	Read words
05	Force Single Coil	0XXXX(bit-output)	%MX0~%MX9999	Write bit
06	Preset Single Register	4XXXX(word-output)	%MW0~%MW9999	Write word
15	Force Multiple Coils	0XXXX(bit-output)	%MX0~%MX9999	Write bits
16	Preset Multiple Registers	4XXXX(word-output)	%MW0~%MW9999	Write words

8) Modbus addressing rules

GM7 base unit starts its address from 0 and matches with 1 of Modicon products' data address. So GM7's address, n matches n+1 of Modicon products' address. Also, GM7 base unit has continuous M area without any division of output contact points (0XXXX), input contact points (1XXXX), output registers (4XXXX), input registers (3XXXX). This means that the output contact point 1 (0001) of Modicon products is marked as communication address 0 and the input contact point 1 (0001) of Modicon products is marked as communication address 0 in GM7.


Output contact points (0XXXX), Input contact points (1XXXX), Output registers (4XXXX), Input registers (3XXXX)
Highest data of data address dividing output contact point, input contact point, output register, and input contact register in Modicon products.

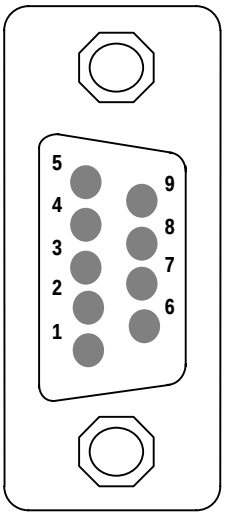
9) The size of using data

As for data size, GM7 base unit supports 128 bytes in ASCII mode and 256 bytes in RTU mode. The maximum size of the Modicon products is different from each other kind. So refer to "Modicon Modbus Protocol Reference Guide."

REMARK

1) GM7 base unit doesn't have any division between input and output area like Modicon PLC, when it supports Modbus protocol communication. It uses only M area. So the user must be caution to in set input and output area in M area for Modbus protocol communication.

10) Map of wiring



Male Type

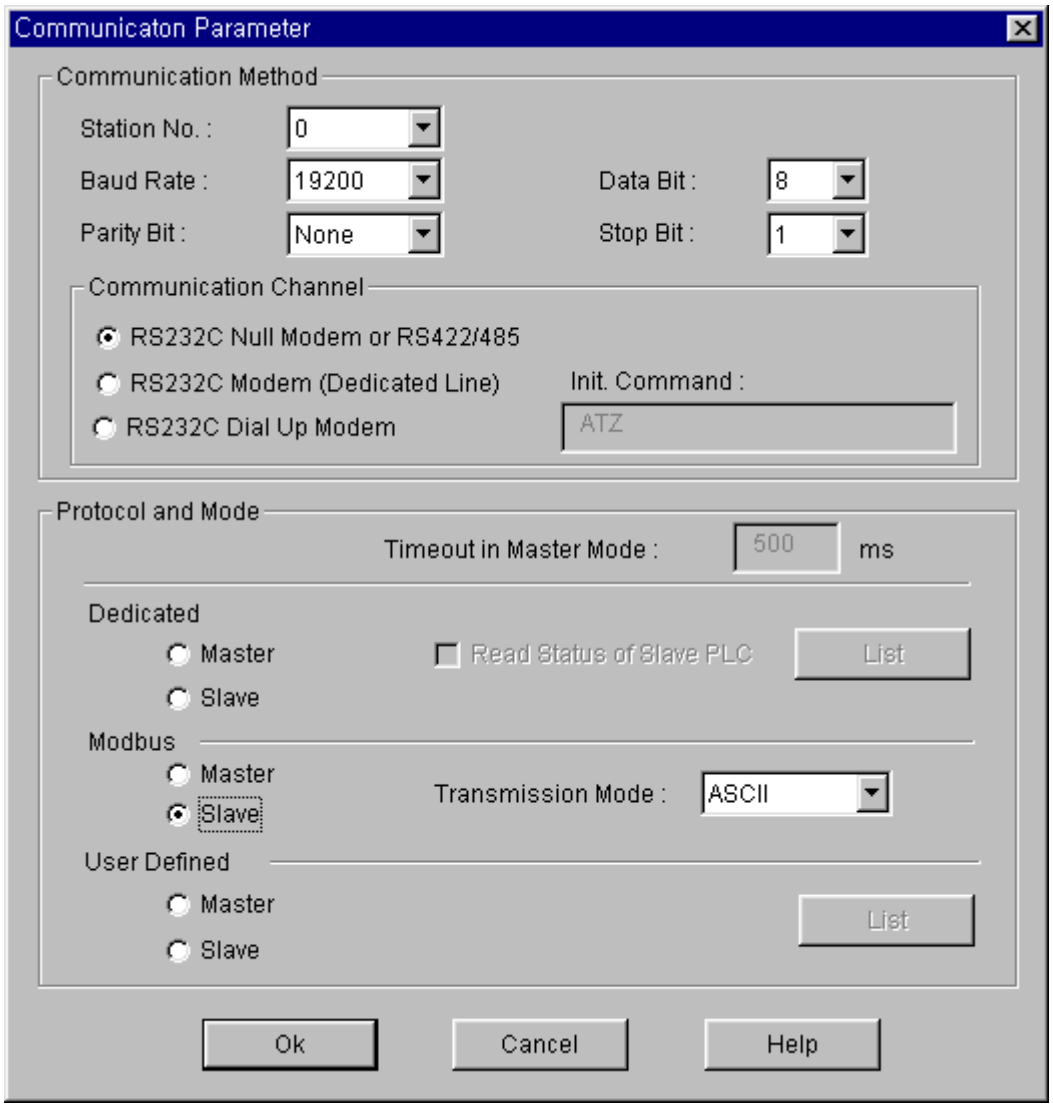
GM7 base unit	Connecting no. and direction	Quantum (9PIN)	
Pin no.		Pin no.	Signal
1		1	CD
2		2	RXD
3		3	TXD
4		4	DTR
5		5	SG
6		6	DSR
7		7	RTS
8		8	CTS
9		9	

8.3.3 Parameters Setting

1) Setting communication parameter

(1) Open a new project file at GMWIN.
- GM7 should be selected in PLC types.
- Open a new project file for each of the master and the slave.

(2) Select a communication parameter at GMWIN and double click to open the following window.

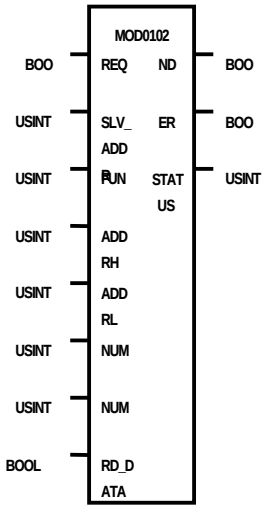


(3) Set the contents as follows.

Item	Setting contents
Station No.	Set a number between 0 to 31 (Don't assign no. 0 as broadcasting station lest it may be a cause for mistaken operation)
Baud Rate	Set one from 1200, 2400, 4800, 9600, 19200, 38400, or 57600 bps.
Data Bit	Set 7 or 8. ASCII mode: Set as 7 bits. RTU mode: Set as 8 bits.
Parity Bit	Set as one of None, Even, or Odd.
Stop Bit	Set 1 or 2 bit(s). When parity bit is set: Set as 1 bit. When parity bit isn't set: Set as 2 bits.
Communication Channel	● RS232C Null Modem or RS422/485 : It's a communication channel for the communication, using GM7 base unit's built-in communication and Cnet I/F module (G7L-CUEC). ● RS232C Modem (Dedicated Line) : It's to be selected for the communication, using an dedicated modem with Cnet I/F module (G7L-CUEB). ● RS232C Dial Up Modem : It's to be selected for the general communication connecting through the telephone line by dial up modem and Cnet I/F module (G7L-CUEB). Footnote) Using Cnet I/F module (G7L-CUEB) supporting RS232C, RS232C dedicated or dial-up modem communication can be done, but not through Cnet I/F module (G7L-CUEC) supporting RS422/485.
Time out in Master Mode	● It's the time waiting a responding frame since the master GM7 base unit sends a request frame. ● The default value is 500ms. ● It must be set in consideration of the max. periodical time for sending/receiving of the master PLC. ● If it's set smaller than the max. send/receive periodical time, it may cause communication error.
Modbus Master/ Slave	If it is set as the master, it's the subject in the communication system. If it's set as the slave, it only responds to the request frame of the master.
Transmission Mode	Select ASCII mode or RTU mode.

8.3.4 Function Block

1) MOD0102

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 01 and 02. ADDRH : High address of the starting addresses to be read from the slave station. ADDRL : Low address of the starting addresses to be read from the slave station. NUMH : High address of the data size of the starting address to be read from the slave station. NUML : Low address of the data size of the starting address to be read from the slave station. **Output** RD_DATA : A variable name to save the data that is read(The number of array is to be declared as same as or bigger than data size.). NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When an error occurs, output an error code.

(1) Function
This is a function block that can execute either function code 01 or 02 for reading bits in Modbus protocol communication. Function code 01 reads Coil Status data and function 02 reads Input Status data.

(2) Error
It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) Example of the program
1 It's supposed that GM7 base unit is the master and it reads Coil Status of the station no. 17, a Modicon product.

- The master reads status of the Coil 00020 ~ 00056 of the slave station no. 17. The Coil of the slave station is supposed to be as follows and the data that are read is saved in any array variable RD_DBD of the 40 sized BOOL type.

Coil	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40
Status	X	X	X	1	1	0	1	1	0	0	0	0	1	1	1	0	1	0	1	1
Hex	1				B				0				E				B			
Coil	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20
Status	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	1	1	0	1
Hex	2				6				B				C				D			

<Data status of the Modicon product's Coil 00020-00059>

- The status of Coil 57, 58, 59 are redundancy.
- Data is sent starting from the low bit by byte unit. If the deficient bit of a byte is filled with 0. An example of sending the above data is as follows.

  Ex1) CD 6B B2 0E 1B

Function block input	Input value	
REQ	Enter the input condition to operate.	
SLV_ADDR	16#11 or 17	: The slave station number.
FUNC	16#01 or 1	: Enter '1' as the Coil Status is being read.
ADDRH	16#00 or 0	: High byte of the starting addresses to be read from the slave station.
ADDRL	16#13 or 19	: High byte of the starting addresses to be read from the slave station. - Read the no. 19 to read starting from the Coil 00020 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0	: High byte of which total data size to be read is expressed in hex.
NUML	16#25 or 37	: Low byte of which total data size to be read is expressed in hex. - Example is to be read 00020 ~ 00056, of which the total data size is 37. 37 are H0025 in hex. So H00 is input for NUHH and H25 for NUHL.

- Results

Variable	Value to save	Variable	Value to save	Variable	Value to save	Variable	Value to save
RD_DB0 [0]	1	RD_DB0 [10]	0	RD_DB0 [20]	1	RD_DB0 [30]	0
RD_DB0 [1]	0	RD_DB0 [11]	1	RD_DB0 [21]	1	RD_DB0 [31]	0
RD_DB0 [2]	1	RD_DB0 [12]	0	RD_DB0 [22]	0	RD_DB0 [32]	1
RD_DB0 [3]	1	RD_DB0 [13]	1	RD_DB0 [23]	1	RD_DB0 [33]	1
RD_DB0 [4]	0	RD_DB0 [14]	1	RD_DB0 [24]	0	RD_DB0 [34]	0
RD_DB0 [5]	0	RD_DB0 [15]	0	RD_DB0 [25]	1	RD_DB0 [35]	1
RD_DB0 [6]	1	RD_DB0 [16]	0	RD_DB0 [26]	1	RD_DB0 [36]	1
RD_DB0 [7]	1	RD_DB0 [17]	1	RD_DB0 [27]	1	RD_DB0 [37]	X
RD_DB0 [8]	1	RD_DB0 [18]	0	RD_DB0 [28]	0	RD_DB0 [38]	X
RD_DB0 [9]	1	RD_DB0 [19]	0	RD_DB0 [29]	0	RD_DB0 [39]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DB0[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

2 It's supposed that GM7 base unit is the master and it reads Input Status of the station no. 17, a Modicon product.

- The master reads status of the Input 10197 ~ 10218 of the slave station no. 17. The Input of the slave station is supposed to be as follows and the data that are previously read is saved in any array variable RD_DB1 of the 25 sized BOOL type.

Input	10220	10219	10218	10217	10216	10215	10214	10213	10212	10211	10210	10209
Status	X	X	1	1	0	1	0	1	1	1	0	1
Hex	3				5				D			
Input	10208	10207	10206	10205	10204	10203	10202	10201	10200	10199	10198	10197
Status	1	0	1	1	1	0	1	0	1	1	0	0
Hex	B				A				C			

- Input coil 10219, 10220 are redundancy.
- Data is sent starting from the low bit by byte unit. If the deficient bit of a byte is filled with 0. An example of sending the above data is as follows.

  Ex1) AC DB 35

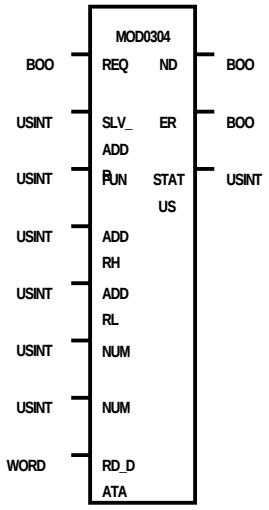
Function block input	Input value	
REQ	Enter the input condition to operate.	
SLV_ADDR	16#11 or 17	: The slave station number.
FUNC	16#02 or 2	: Enter '2' as the Input Status is being read.
ADDRH	16#00 or 0	: High byte of the starting addresses to be read from the slave station.
ADDRL	16#C4 or 196	: High byte of the starting addresses to be read from the slave station. - Read the no. 196 to read starting from the input coil 10197 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0	: High byte of which total data size to be read is expressed in hex.
NUML	16#16 or 22	: Low byte of which total data size to be read is expressed in hex. - Example is to read 10197-10218, of which the total data size is 22. 22 are H0016 in hex. So H00 is input for NUHH and H16 for NUHL.

- Results

Variable	Value to save	Variable	Value to save	Variable	Value to save	Variable	Value to save
RD_DB1 [0]	0	RD_DB1 [6]	0	RD_DB1 [12]	1	RD_DB1 [18]	1
RD_DB1 [1]	0	RD_DB1 [7]	1	RD_DB1 [13]	0	RD_DB1 [19]	0
RD_DB1 [2]	1	RD_DB1 [8]	1	RD_DB1 [14]	1	RD_DB1 [20]	1
RD_DB1 [3]	1	RD_DB1 [9]	1	RD_DB1 [15]	1	RD_DB1 [21]	1
RD_DB1 [4]	0	RD_DB1 [10]	0	RD_DB1 [16]	1	RD_DB1 [22]	X
RD_DB1 [5]	1	RD_DB1 [11]	1	RD_DB1 [17]	0	RD_DB1 [23]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DB1[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

2) MOD0304

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 03 and 04. ADDRH : High address of the starting addresses to be read from the slave station. ADDRL : Low address of the starting addresses to be read from the slave station. NUMH : High address of the data size of the starting address to be read from the slave station. NUML : Low address of the data size of the starting address to be read from the slave station. **Output** RD_DATA : A variable name to save the data that is read(The number of array is to be declared as same as or bigger in data size.). NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When an error occurs, output an error code.

(1) Function

This is a function block that can execute either function code 03 or 04 for reading words in Modbus protocol communication. Function code 03 reads Holding Registers and function 04 reads Input Registers.

(2) Error

It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) Example of the program

1 It's supposed that GM7 base unit is the master and it reads Holding Registers of the station no. 17, a Modicon product.

- The master reads the Holding Registers 40108 ~ 40110 of the slave station no. 17. The status of the Holding Registers of the slave station is supposed to be as follows and the previously read data are saved in any array variable RD_DWO of the 40 sized WORD type.

Holding Registers	40110	40109	40108
Register status	H0064	H0000	H022B

- Data is sent starting from the low bit by byte unit. An example of sending the above data is as follows.

  Ex1) 02 2B 00 00 00 64

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#03 or 3 : Enter '3' as the Holding Registers is being read.
ADDRH	16#00 or 0 : High byte of the starting addresses to be read from the slave station.
ADDRL	16#6B or 107 : High byte of the starting addresses to be read from the slave station. - Read the no. 107 to read starting from the output holding Registers 40108 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte of which total data size to be read is expressed in hex.
NUML	16#03 or 3 : Low byte of which total data size to be read is expressed in hex. - Example is to read 40108 ~ 40110, of which the total data size is 3. 3 are H0003 in hex. So H00 is input for NUHH and H03 for NUHL.

- Result

Variable	Value to save
RD_DW0 [0]	H002B or 555
RD_DW0 [1]	H0000 or 0
RD_DW0 [2]	H0064 or 100
RD_DW0 [3]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DW0[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

2 It's supposed that GM7 base unit is the master and it reads output coil data of the station no. 17, a Modicon product.

- The master reads status of the input registers 30009 of the slave station no. 17. The input coil of the slave station is supposed to be as follows and the data that are read is saved in any array variable RD_DW1 of the 2-sized WORD type.

Input Register	30009
Register status	H000A

- Data is sent starting from the low bit by byte unit. An example of sending the above data is as follows.

  Ex1) 00 0A

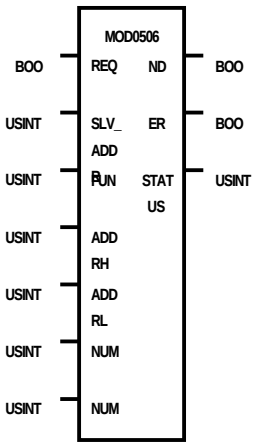
Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#04 or 4 : Enter '4' as the Input Registers is being read.
ADDRH	16#00 or 0 : High byte of the starting addresses to be read from the slave station.
ADDRL	16#08 or 8 : High byte of the starting addresses to be read from the slave station. - Read the no. 8 to read starting from the input Registers 30009 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte of which total data size to be read is expressed in hex.
NUML	16#01 or 1 : Low byte of which total data size to be read is expressed in hex. - Example is to read only 30009, of which the total data size is 1. 1 is H0001 in hex. So H00 is input for NUHH and H01 for NUHL.

- Results

Variable	Value to save
RD_DW1 [0]	H000A or 10
RD_DW1 [1]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DW1[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

3) MOD0506

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 05 and 06. ADDRH : High address of the starting addresses to write on the slave station. ADDRL : Low address of the starting addresses to write on the slave station. NUMH : High address of the data size of the starting address to write on the slave station. NUML : Low address of the data size of the starting address to write on the slave station. **Output** NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When error occurs, output an error code.

(1) Function

This is a function block that can execute either function code 05 or 06 for writing 1 bit (function code 05) and writing 1 word (function code 06) in Modbus protocol communication. Function code 05 does 1 bit data writing on the Output Coil. If the Input NUMH is set as 255 (or HFF), it writes 1 on the output coil. If the Input NUMH is set as 0 (or H00), it writes 0 on the output coil. And function 06 does 1 word data writing on the Output Holding Register.

(2) Error

It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) An example of the program

1 It's supposed that GM7 base unit is the master and it writes 1 bit data on the Coil of the station no. 17, a Modicon product.

2 The master writes 1 on the Coil 00173 of the slave station no. 17, a Modicon product.

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#05 or 5 : Enter '5' as writes bit by bit on the Coil.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#AC or 172 : High byte of the starting addresses to write on the slave station. - Write on the no. 172 to write on, starting from the output coil 00173 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#FF or 255 : The data to be written on the slave station. - '0' H00 - '1' HFF
NUML	16#00 or 0 : The data to be written on the slave station. - As the example writes 1, enter HFF for HUMH and H00 for NUML

- Result : The Coil 00173 turns ON. (In case of GM7 base unit, 1 is saved on the related M area.)

Coil	00173
Status	1

3 It's supposed that GM7 base unit is the master and it writes on 1 words Holding Registers of the station no. 17, a Modicon product.

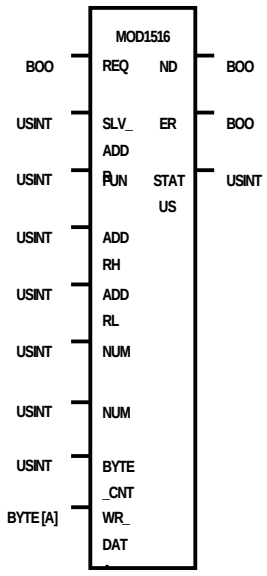
- An example of writing 3 on Holding Register 4002 of the station no. 17.

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#06 or 6 : Enter '6' as 1 word is being written on the Holding Register.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#01 or 1 : High byte of the starting addresses to write on the slave station. - Write on no. 1 to write on, starting from the Holding Register 40002 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte when the data to be written on is expressed in hex.
NUML	16#03 or 3 : High byte when the data to be written on is expressed in hex. - As the example writes 3, of which hex is H0003. So H00 is input for NUHH and H03 for NUHL.

- Result : The Holding Register 40002 is saved on H0003. (In case of GM7 base unit, H0003 is saved on a related M area.)

Holding Register	40002
Register status	H0003

4) MOD1516

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 15 and 16. ADDRH : High address of the starting addresses to write on the slave station. ADDRL : Low address of the starting addresses to write on the slave station. NUMH : High address of the data size of the starting address to write on the slave station. NUML : Low address of the data size of the starting address to write on the slave station. BYTE_CNT : The size of data to be written on the slave station. WR_DATA : Variable name to save the data to be written(The number of array is to be declared as same as or bigger in data size.). **Output** NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When an error occurs, output an error code.

(1) Function

This is a function block that can execute either function code 15 or 16 for writing 1 bit (function code 15) and writing 1 word (function code 16) in Modbus protocol communication. Function code 15 does 1 bit by 1 bit data writing on each Coil in a sequence Coils. And function 16 does 1 word by 1 word data writing on sequence of the Holding Registers.

(2) Error

It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) An example of the program

1 It's supposed that GM7 base unit is the master and it writes bits continually on the output coil of the station no. 17, a Modicon product.

- The master writes continual 10 bits, 01110011011 on the Coils 00020 of the slave station no. 17 1 bit by 1 bit. The data that is to be written are saved in any array variable WR_DB0 of the 2 sized BYTE type.

Variable	Value to save
WR_DB0 [0]	2#11001101 16#CD
WR_DB0 [1]	2#10000001 16#81

- The size of BYTE_CNT is the same as when the data to be written are converted by byte. The above data are 10 by 1 bit. They can't be filled by 1 byte. So they must be filled from the low bit, using 2 bytes. And 0 fills the remnant 6 bits. Therefore the size of BYTE_CNT is 2.
- If it is supposed that data of 1000 0001 1100 1101 are saved in the array variable, WR_DB0, the data are sent as 10 bits (01 1100 1101) at the bottom plus 6 bit of 0 at the top. For the size of the data is set as 10 bits to send and they are sent by bytes, the deficient 6 bits are filled with 0.
- Data is sent starting from the low bit by byte unit. An example of sending the above data is as follows.

  Ex1) CD 01

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#0F or 15 : Enter '15' as bits are continually written on the output coils.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#13 or 19 : Low byte of the starting addresses to write on the slave station. - Write on No. 19 to write on, starting from the output holding Registers 00020 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte when the data to be written on is expressed in hex.
NUML	16#0A or 10 : Low byte when the data to be written on is expressed in hex. - For the data of the example are 10 bits continued from 00020, their size is 10, which is H000A in hex. So H00 is input for NUHH and H0A for NUHL.
BYTE_CNT	16#02 or 2 : The size of the data to be written on the slave station, when they are converted by bytes. - For the data of the example are continual 10 bits, which is converted to 2 bytes. Therefore input H02 for BYTE_CNT.

- Result
  From the 2 bytes (16 bits) sent, only the low 10 bits are valid as set for its size.

Coil	00029	00028	00027	00026	00025	00024	00023	00022	00021	00020
Status	0	1	1	1	0	0	1	1	0	1

2 It's supposed that GM7 base unit is the master and it writes word data continually on the Holding Registers of the station no. 17, a Modicon product.

- The master writes 000A and 0102 on the Holding Registers 40002 of the slave station no. 17. The data that is to be written are saved in any array variable WR_DB1 of the 4 sized BYTE type.

Variable	Value to save
WR_DB1 [0]	2#00001010 or 16#0A
WR_DB1 [1]	2#00000000 or 16#00
WR_DB1 [2]	2#00000010 or 16#02
WR_DB1 [3]	2#00000001 or 16#01

- The size of BYTE_CNT is the same as when the data to be written are converted by byte. The above data are 2 words that need 4 bytes. Therefore the size of BYTE_CNT is 4.
- Data is sent starting from the low word by byte unit. An example of sending the above data is as follows.

  Ex1) 00 0A 01 02

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#10 or 16 : Enter '16' as words are continually written on the Holding Registers.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#01 or 1 : Low byte of the starting addresses to write on the slave station. - Write on no. 1 to write on, starting from the output holding Registers 40002 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte when the data to be written on is expressed in hex.
NUML	16#02 or 2 : Low byte when the data to be written on is expressed in hex. - For the data of the example are 2-word data continued from 40002, their size is 2, which is H0002 in hex. So H00 is input for NUHH and H02 for NUHL.
BYTE_CNT	16#04 or 4 : The size of the data to be written on the slave station, when they are converted by bytes. - For the data of the example are continual 2-word data, which is converted to 4 bytes. Therefore input H04 for BYTE_CNT.

- Result

Holding Registers	40003	40002
Registers status	H0102	H000A

5) Error code

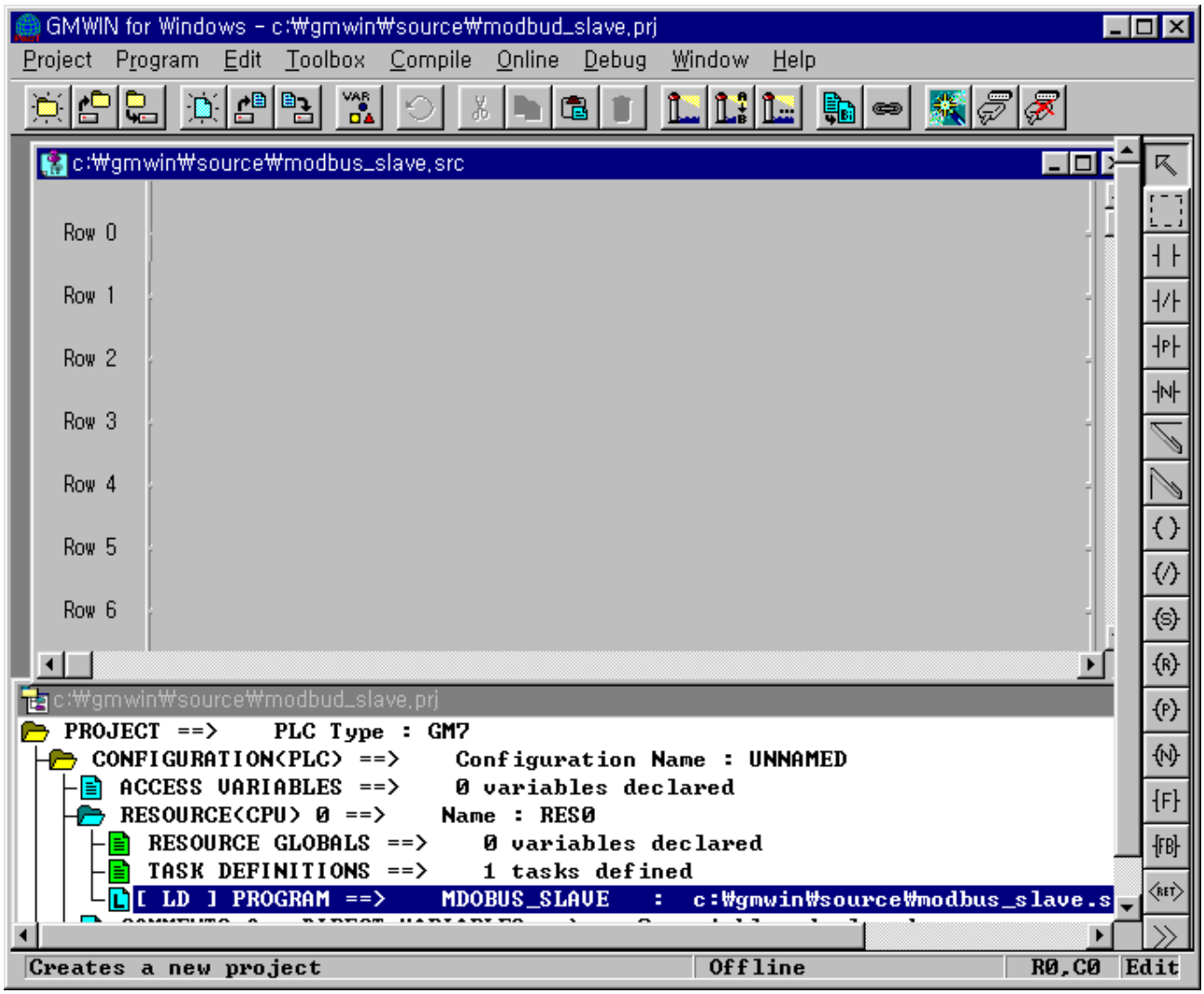
CODE	Error type	Meaning
01	Illegal Function	Error in inputting function code in function block.
02	Illegal Address	Error of exceeding the area limit of reading/writing on the slave station.
03	Illegal Data Value	Error when the data value to be read from or write on the slave station isn't allowed.
04	Slave Device Failure	Error status of the slave station.
05	Acknowledge	It's a responding code of the slave station for the master station to prevent the master stations time-out error, when request command processing takes time. The master station marks an error code and waits for a certain time without making any second request.
06	Slave Device Busy	Error when request command processing takes too much time. The master should request again.
07	Time Out	Error when exceeds the time limit of the communication parameter as it communicates.
08	Number Error	Errors when data is 0 or more than 256 bytes, when the data size is bigger than the array size, or when Number and BYTE_CNT are different from each other.
09	Parameter Error	Error of setting parameters (mode, master/ slave)
10	Station Error	Error when the station number of itself and the station number set by the input parameter of the function block are the same.

8.3.5 Example of Use

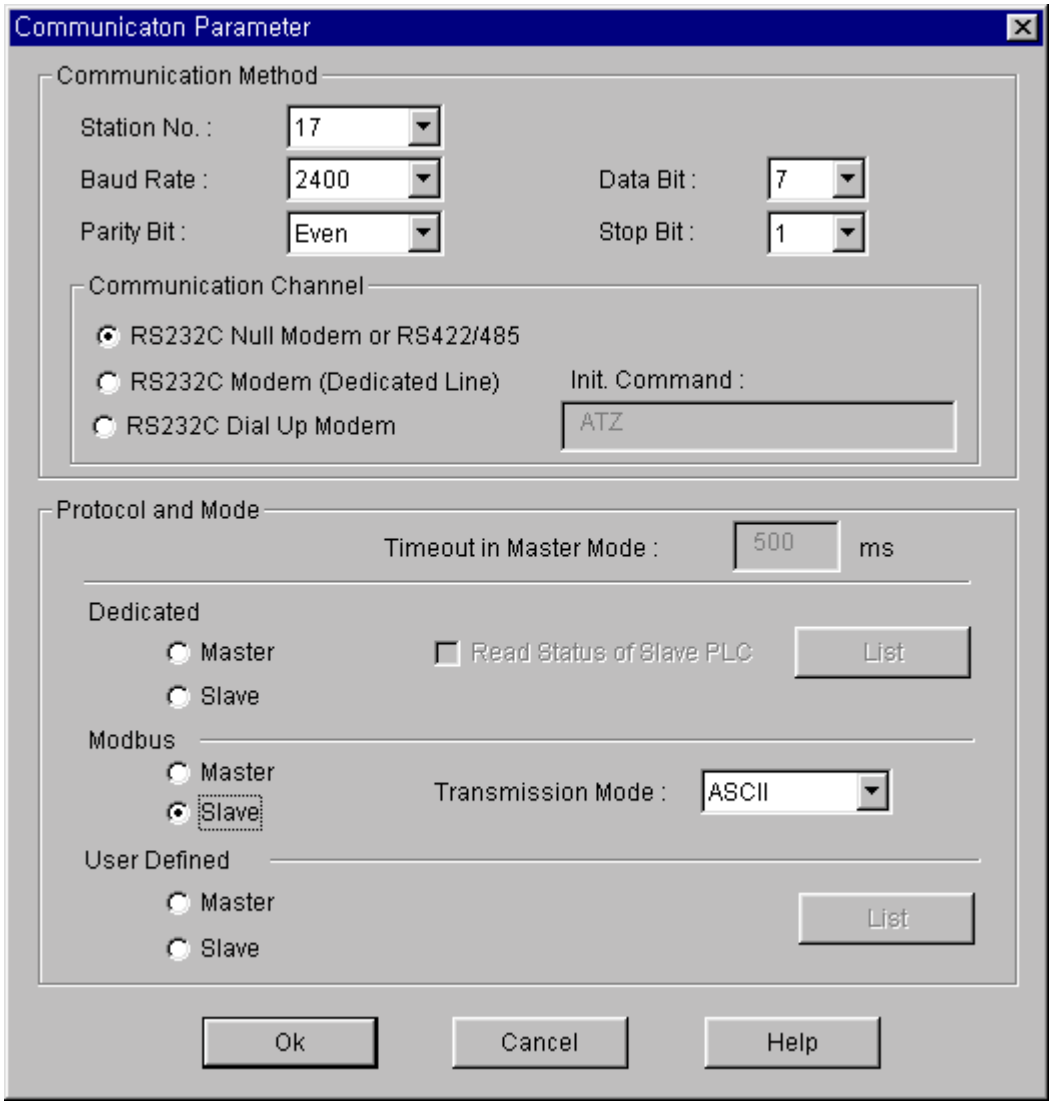
- According to the setting of device supporting Modbus protocol, setting of GM7 basic unit is changed, but this example explains Modbus protocol communication among GM7 units.
- The slave station program: This outputs the received data saved in M area through the output coil.
- The master station program : It saves 16#FF (or 255) at %MW0 (It is coincided with %MX0 ~ MX15 or %MB0 ~ %MB1) in function block MOD0506 (function code 06), then reads %MX0 through MOD0102 (function code 01), and again saves 0 at %MX0 ~ %MX9 using function block MOD1516 (function code 15), then reads %MW0 through Mod0304.
- The cable used in this example is same with that used for the dedicated protocol communication between GM7's.

1) Setting and the program of the slave station

(1) Open a new project file and a new program for the slave station.



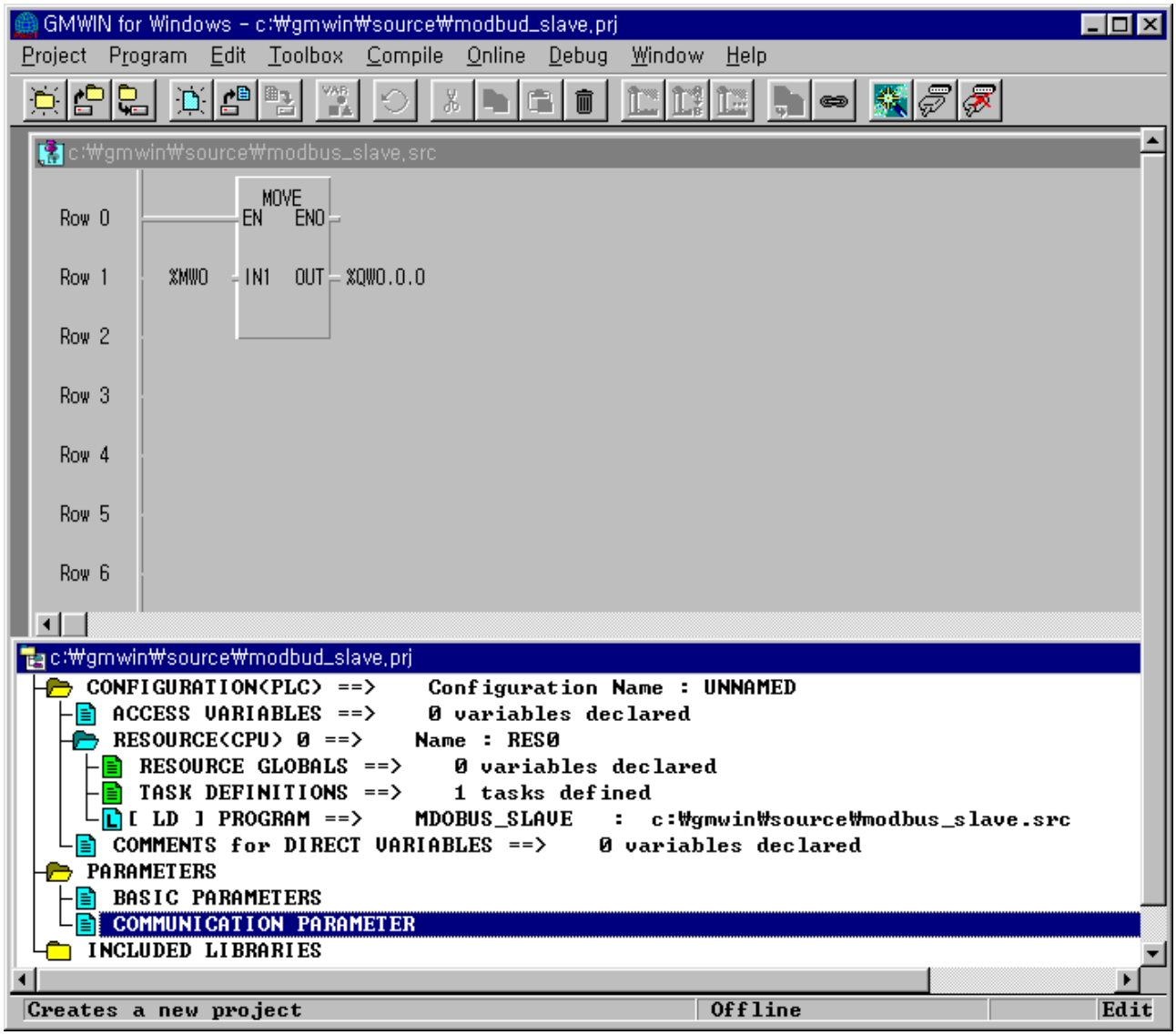
(2) Select communication parameter in GMWIN parameter and double-click on it, the window of communication parameter opens.



- Set parameters as the following table.

Communication Method						Protocol and Mode	
Station No.	Baud Rate	Data Bit	Parity Bit	Stop Bit	Communication Channel	Modbus	Transmission Mode
17	2400	7	Even	1	RS232C Null Modem or RS422/485	Slave	ASCII

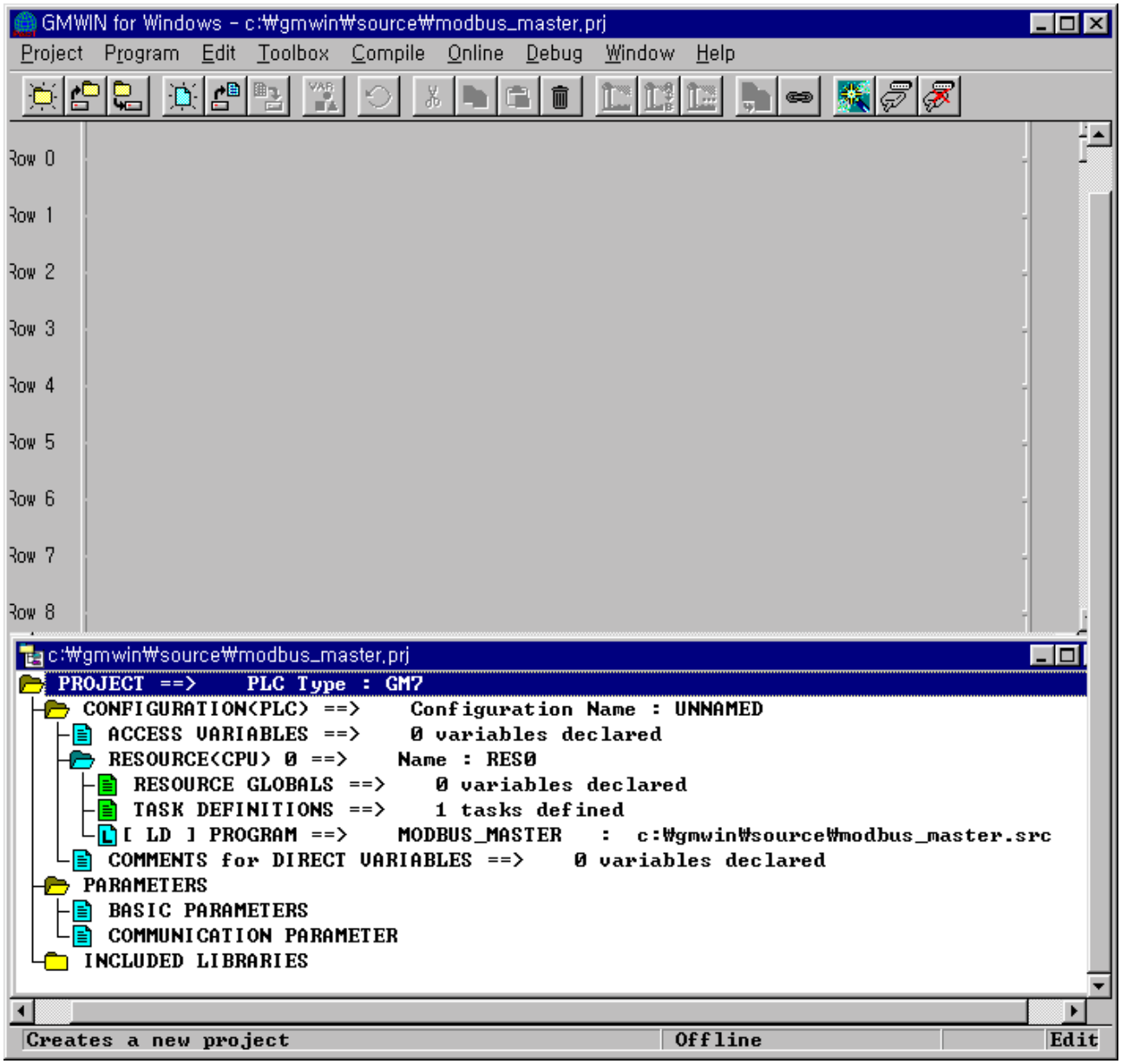
(3) Set up a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.



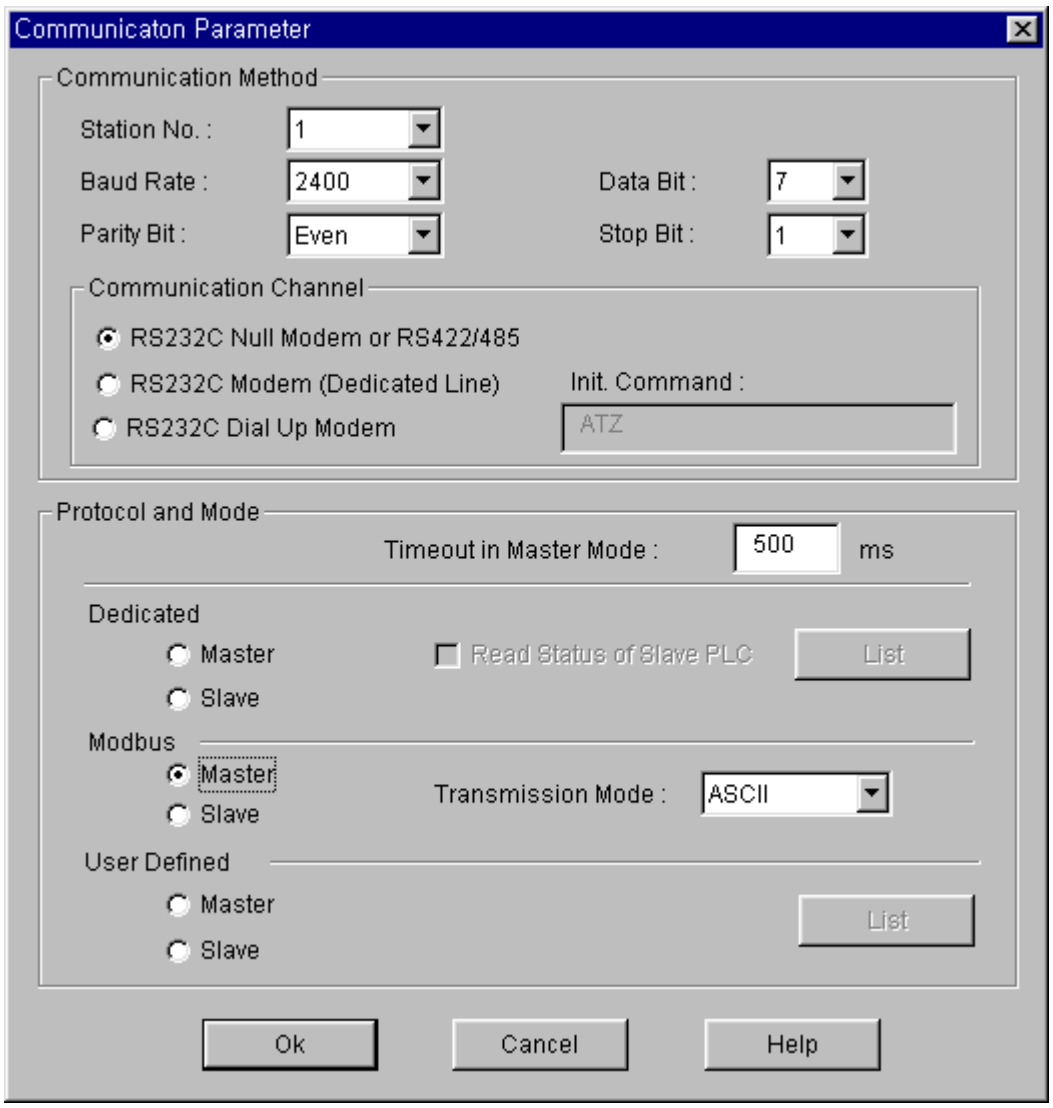
- The program of the slave is to output the data at M area to the output contact coil.

2) Setting and the program of the master station

(1) Make a new project file and a new program for the master station.



(2) If you double click the 'communication parameter' window in GMWIN, you can see the following window of the 'communication parameter.'

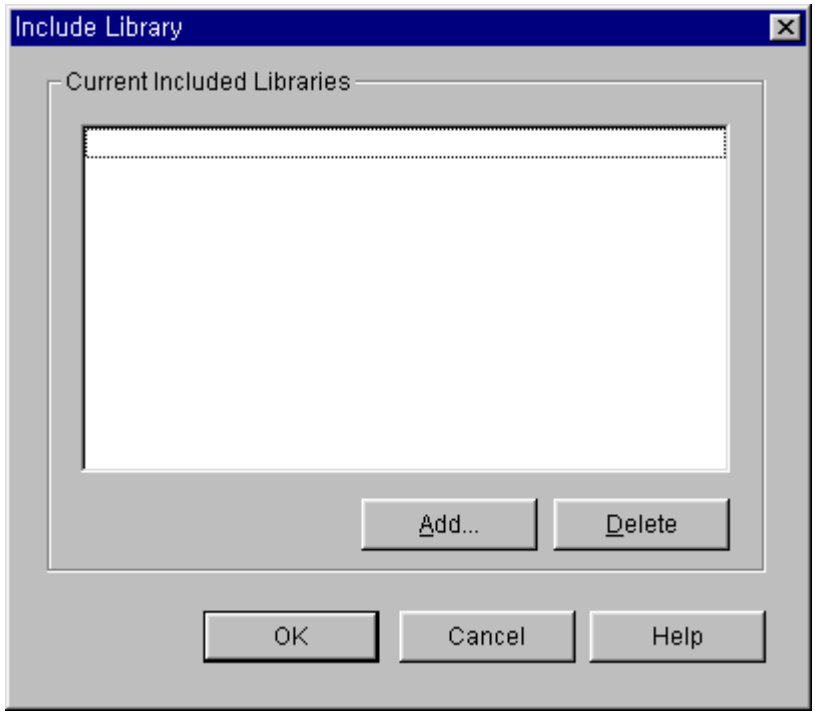


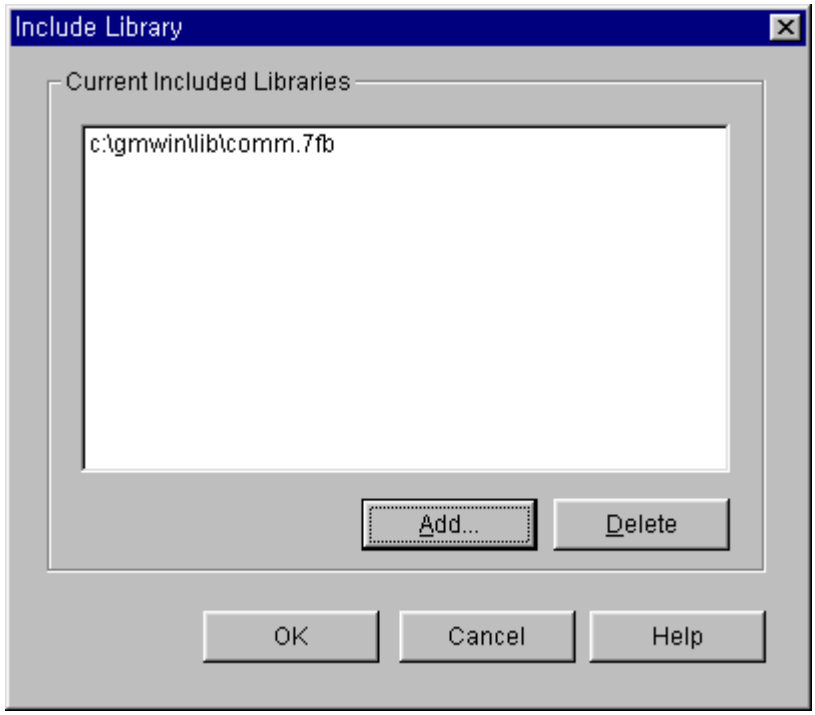
- Set parameters as the following table.

Communication Method						Protocol and Mode	
Station No.	Baud Rate	Data Bit	Parity Bit	Stop Bit	Communication Channel	Modbus	Transmission Mode
1	2400	7	Even	1	RS232C Null Modem or RS422/485	Master	ASCII

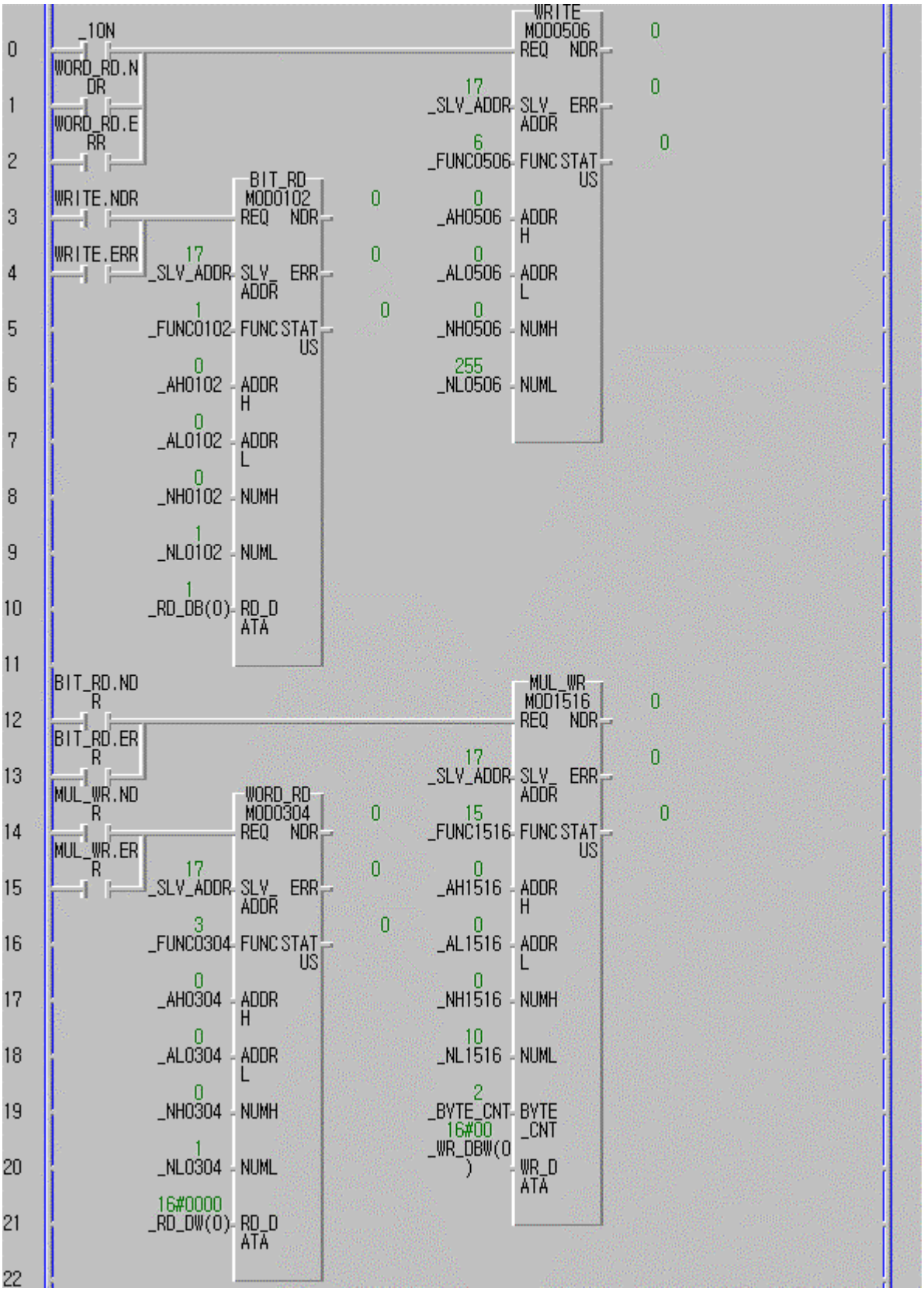
(3) Set up a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.

- Function block is used in the program. Before using function block, double click “Current included Libraries” to open the following window. Click “Add” to add COMM.7FB and click “OK.”





- Framing program



- It saves 16#FF (or 255) at %MW0 (It is coincided with %MX0 ~ %MX15 or %MB0 ~ %MB1) in function block MOD0506 (function code 06), then reads %MX0 through MOD0102 (function code 01), and again saves 0 at %MX0 ~ %MX9 using function block MOD1516 (function code 15), then reads %MW0 through Mod0304.
- Then you will see that 8 LEDs of output contact point operate on/off continually.
- The above picture is the monitored scene of the program operation. Therefore the values appeared in _RD_DB, _RD_DW, array variables are not the initial ones, but the resulted value after executing reading.
- Variables like Instanse name.NDR, Instanse name.ERR, Instanse name.STATUS are automatically generated when an instanse variable of function block is declared.
- _1ON flag is a flag that is on for 1 scan.
- Each function block input REQ conditions to make separate function block output as its input.
- The size of _BYTE_CNT must be the same when it is converted into bytes.
- Error occurs when the size of array variable is smaller than the data to be read or to be written.

- Table of variables

Variable	Variable type	Initial value	Variable	Variable type	Initial value
_SLV_ADDR	USINT	17(H11)	_NH0102	USINT	0(H00)
_FUNC0102	USINT	1(H01)	_NH0304	USINT	0(H00)
_FUNC0304	USINT	3(H03)	_NH0506	USINT	0(H00)
_FUNC0506	USINT	6(H06)	_NH1516	USINT	0(H00)
_FUNC1516	USINT	15(H0F)	_NL0102	USINT	1(H01)
_AH0102	USINT	0(H00)	_NL0304	USINT	255(HFF)
_AH0304	USINT	0(H00)	_NL0506	USINT	1(H01)
_AH0506	USINT	0(H00)	_NL1516	USINT	10(H0A)
_AH1516	USINT	0(H00)	_RD_DB	BOOL-typed ARRAY [40]	{0,0,0}
_AL0102	USINT	0(H00)	_RD_DW	WORD-typed ARRAY [4]	{0,0,0,0}
_AL0304	USINT	0(H00)	_WR_DBW	BYTE-typed ARRAY [4]	{0,0,0,0}
_AL0506	USINT	0(H00)	_BYTE_CNT	USINT	2(H02)
_AL1516	USINT	0(H00)			

Chapter 9. INSTALLATION AND WIRING

9.1 Installation

9.1.1 Installation Environment

This unit has high reliability regardless of its installation environment, but be sure to check the following for system reliability.

1) Environment requirements

Avoid installing this unit in locations which are subjected or exposed to:

(1) Water leakage and dust.
(2) Continuous shocks or vibrations.
(3) Direct sunlight.
(4) Dew condensation due to rapid temperature change.
(5) Higher or lower temperatures outside the range of 0 to 55□
(6) Relative humidity outside the range of 5 to 95□
(7) Corrosive or flammable gases

2) Precautions during installing

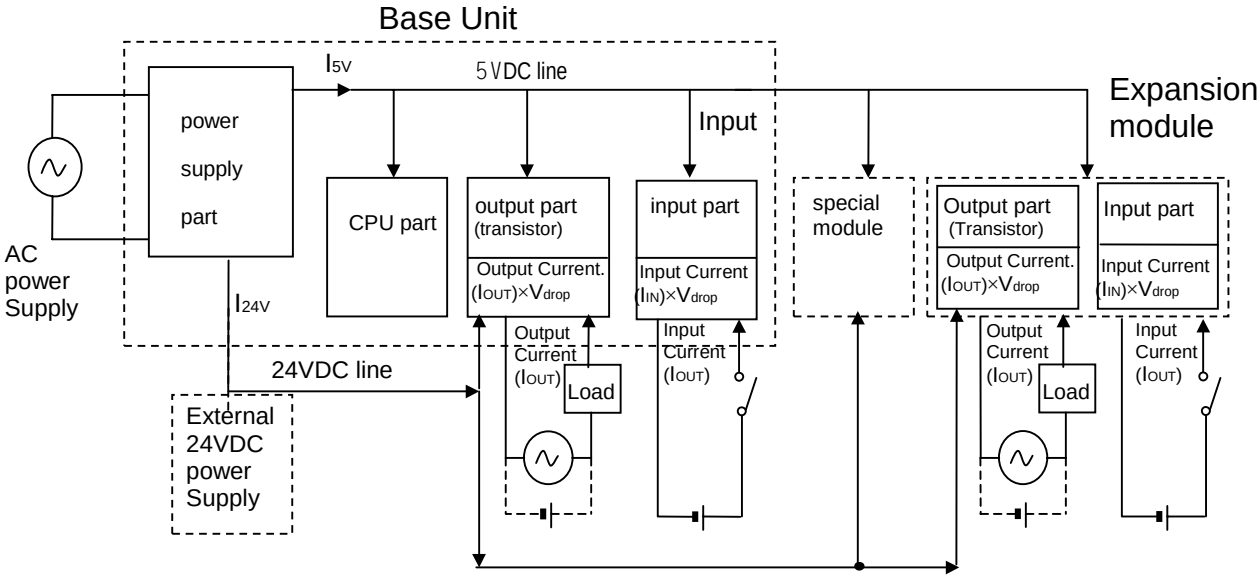
(1) During drilling or wiring, do not allow any wire scraps to enter into the PLC.
(2) Install it on locations that are convenient for operation.
(3) Make sure that it is not located on the same panel that high voltage equipment located.
(4) Make sure that the distance from the walls of duct and external equipment be 50mm or more.
(5) Be sure to be grounded to locations that have good ambient noise immunity.

3) Heat protection design of control box

(1) When installing the PLC in a closed control box, be sure too design heat protection of control box with consideration of the heat generated by the PLC itself and other devices.
(2) It is recommended that filters or closed heat exchangers be used.

The follwing shows the procedure for calculating the PLC system power consumption.

1) PLC system power consumption block diagram



2) Power consumption of each part

(1)Power consumption of a power supply part

Approximately 70% of the power supply module current is converted into power 35% of that 65% dissipated as heat, i.e., 3.5/6.5 of the output power is actually used.

- $Wpw = 3.5 / 6.5 \{(I_{5V} \times 5) + (I_{24V} \times 24)\}$ (W)

where, I5v: 5VDC circuit current consumption of each part

I24v: 24VDC circuit average current consumption of output part (with points simultaneously switched ON). Not for 24VDC power supplied from external or power supply part that has no 24VDC output.

(2) Total 5VDC power consumption

The total power consumption of all modules is the power of the 5VDC output circuit of the power supply part.

- $W_{5V} = I_{5V} \times 5$ (W)

(3) Average DC24V power consumption (with points simultaneously switched ON)

The tatal power consumption of all modules is the average power of the DC24V output circuit of the power

supply part.

- $W_{24V} = I_{24V} \times 24$ (W)

(4) Average power consumption by voltage drop of output part (with points simultaneously switched ON)

- $W_{out} = I_{out} \times V_{drop} \times$ output points $\times$ the rate of points switched on simultaneously (W)

I_{out} : output current (actual operating current) (A)

V_{drop} : voltage dropped across each output load (V)

(5) Average power consumption of input circuits if input part (with points simultaneously ON)

- $W_{in} = I_{in} \times E \times$ input points $\times$ the rate of points switched on simultaneously (W)

  I_{in} : input current (effective value for AC) (A)

  E : input voltage (actual operating voltage) (V)

(6) Power consumption of the special module

- $W_S = I_{5V} \times 5 + I_{24V} \times 24$ (W)

The sum of the above values is the power consumption of the entire PLC system.

- $W = W_{PW} + W_{5V} + W_{24V} + W_{out} + W_{in} + W_s$ (W)

Check the temperature rise within the control panel with calculation of that total power consumption(W). The temperature rise in the control panel is expressed as:

$T = W / UA$ [°C]

W : Power comsumption of the entire PLC system(obtained as shown above)

A : Control panel inside surface area [m^2]

U : if the control panel temperature is controlled by a fan, etc 6

if control panel air is not circulated 4

9.1.2 Handling Instructions

- Do not drop it off, and make sure that strong shock should not be applied.
- Do not unload the PCB from its case. It can cause faults.
- During wiring, be sure to check any foreign matter like wire scraps should not enter into the upper side of the PLC.

If any foreign matter has entered into it always eliminate it.

1) Base unit or Expansion Module handling instructions

The followings explains instructions for handling or installing the Base unit or Expansion Module.

(1) I/O specifications re-check

Re-check the input voltage for the input part. if a voltage over the maximum switching capacity is applied, it can cause faults, destruction or fire.

(2) Used wire

Select the wire with due consideration of ambient temperature and rated current.
Its minimum specifications should be AWG24(0.18) or more.

(3) Environment

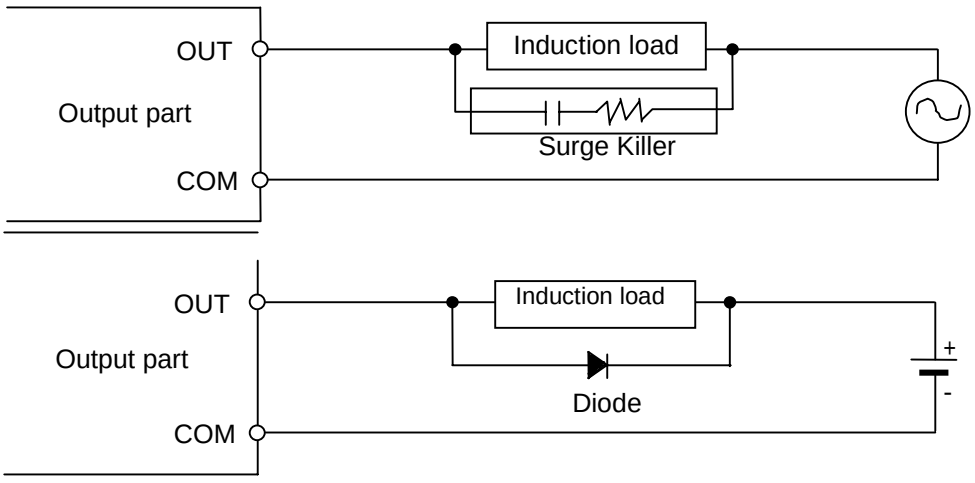
When wiring the I/O part, if it locates near a device generating an cause short circuit, destruction or malfunction.

(4) Polarity

Before applying the power to part that has polarities, be sure to check its polarities.

(5) Wiring

- Wiring I/O wires with high voltage cable or power supply line can cause malfunction or disorder.
- Be sure that any wire does not pass across during input LED(I/O status will not be clearly identified).
- If an inductive load has been connected to output part, connect parallel surge killer or diode to a load.Connect the cathode part of the + part of the power supply.



(6) Terminal block

Check its fixing. During drilling or wiring, do not allow any wire scraps to enter the PLC.
It can cause malfunction and fault.

(7) Be cautious that strong shock does not applied to the I/O part.

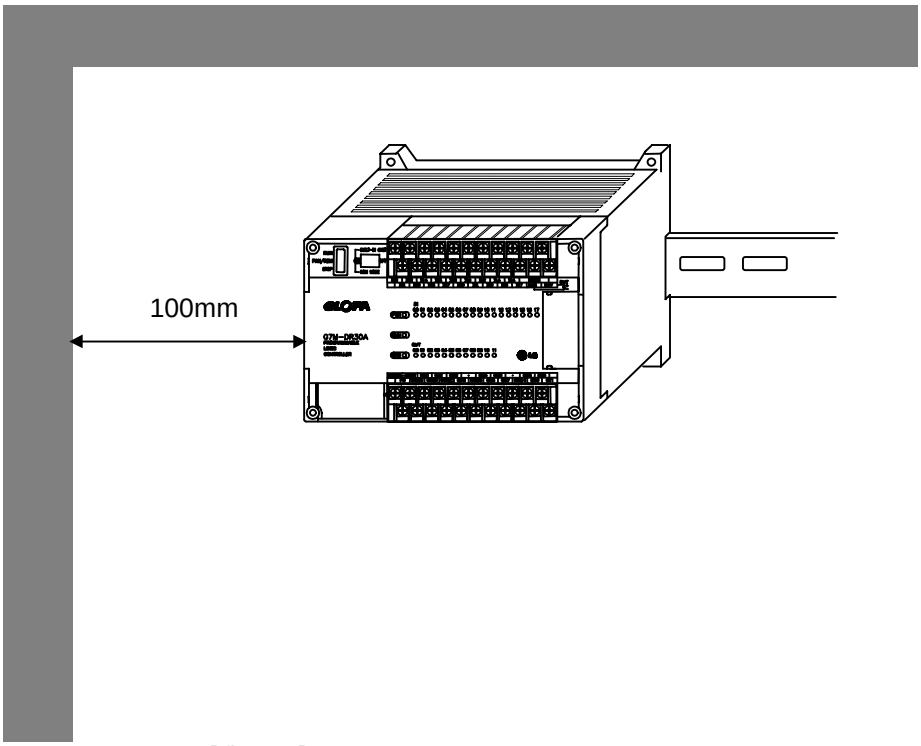
Do not separate the PCB from its case.

2) Mounting instructions

The following explains instructions for mounting the PLC onto the control panel.

(1) Allow sufficient distance from upper part of the Unit for easy module replacement and ventilation. Especially the distance between the left side of the basic unit and the control panel should be 100 or more for periodic battery replacement.

(2) Make sure that GM7 is installed in fig 9.1 for most effective heat radiation.



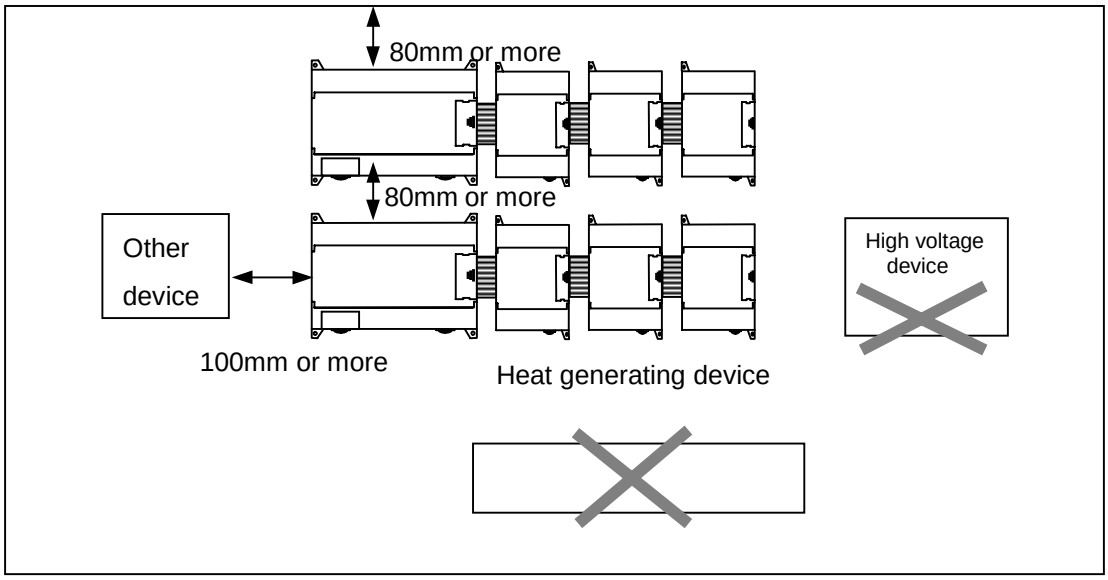
[fig 9.1]

(3)Do not mount the base board together with a large-sized electromagnetic contact or no-fuse breaker, which produces vibration, on the same panel. Mount them on different panels, or keep the unit or module away from such a vibration source

(4) Mount the wire duct as it is needed.
If the clearances are less than those in Fig 9.1, follow the instructions shown below

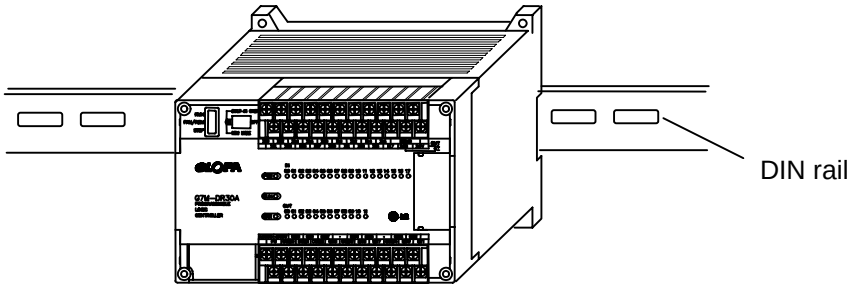
- If the wire duct is mounted on the upper part of the PLC, make the wiring duct clearance 50 or less for good ventilation. Also, allow the distance enough to press the hook in the upper part from the upper part of the PLC.
- If the wire duct is mounted on the lower part of the PLC, make optic or coaxial cables contact it and consider the minimum diameter of the cable.

(5) To protect the PLC from radiating noise or heat, allow 100 or more clearances between it and parts. Left or right clearance and clearance from other device in the left or right side should be 100 or more.



[Fig 9.2] PLC mounting

(6) GM7 has hooks for DIN rail in the base unit and expansion modules.

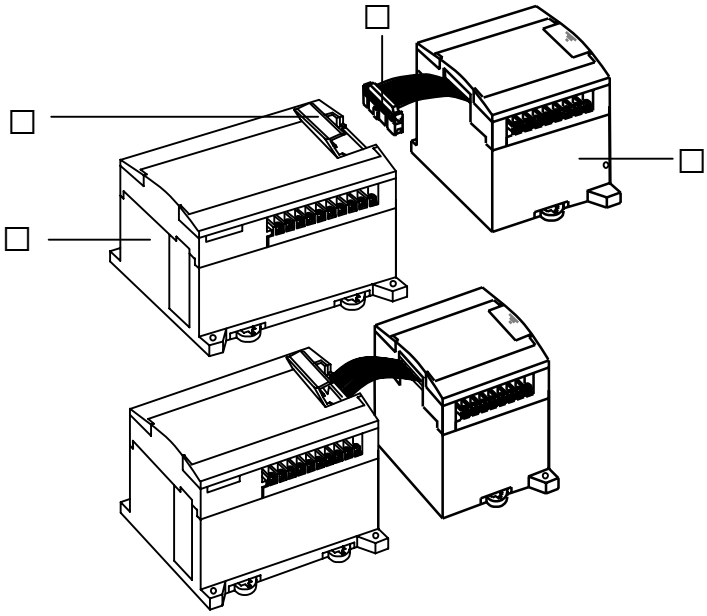


[Fig 10.3]

9.1.3 Connection of expansion module

The following explains the Connection of expansion modules to the base unit.

(1) Open the connector cover of the base unit.
(2) Insert the connector of the expansion module to the connector of the base unit.



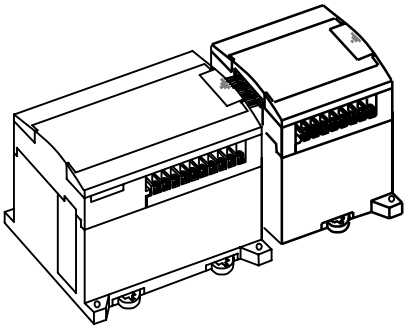
□: Base unit

□: Connector cover

□: expansion module

□: expansion cable

（3）Close the connector cover of the base unit.



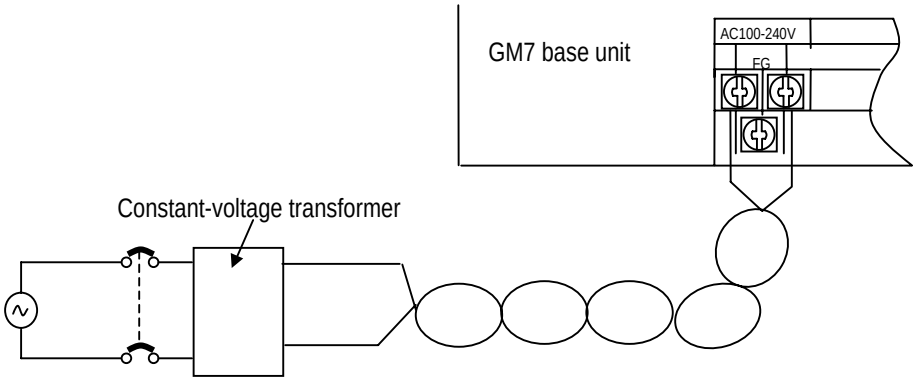
9.2 Wiring

The followings explains the wiring instructions for use of the system.

9.2.1 Power Supply Wiring

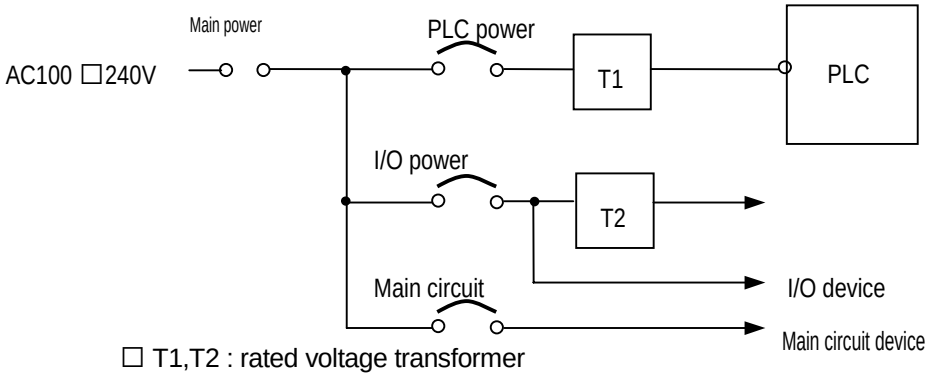
1)Use AC 100 □240V (50Hz□60Hz) as the main power.

2)When voltage fluctuations are larger than the specified value, connect a constant-voltage transformer.
Use a power supply which generates minimal noise across wire and GM7 and ground when excessive noise
Generated, connect an insulating transformer.



3)Use a power supply which generates minimal noise across wire and across PLC and ground. (When excessive noise is generated, connect an insulating transformer)

4) When wiring, separate the PLC power supply from those for I/O and power device as shown below.

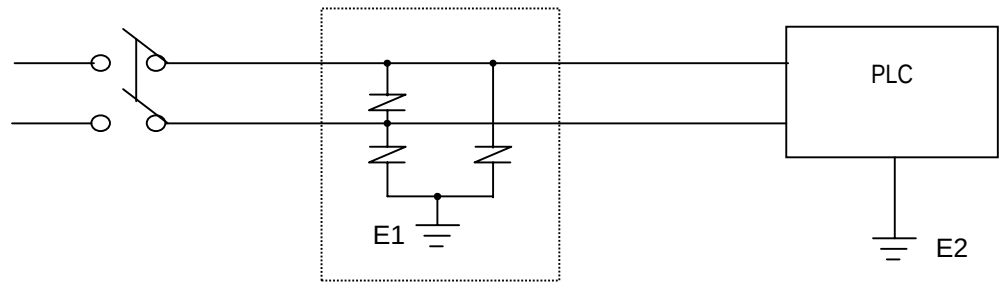


□ T1,T2 : rated voltage transformer

5) To minimize voltage drop, use the thickest (max. 2□) wires possible

6) Do not bundle the 100 VAC and 24VDC cables with main-circuit (high voltage, large current) wires or the I/O signal wires. If possible, provide more than 80 distance between the cables and wires.

7) As a measure against verylarge surge(e.g. due to lightening),connect a surge absorber as shown below.



Surge absorber for lightening

Remark

1) Ground the surge absorber(E1) and the PLC(E2) separately from each other.
2) Select a surge absorber making allowances for power voltage rises.

8) Use a insulating transformer or noise filter for protection against noise.

9) Twist every input power supply wires as closely as possible. Do not allow the transformer or noise filter across the duct.

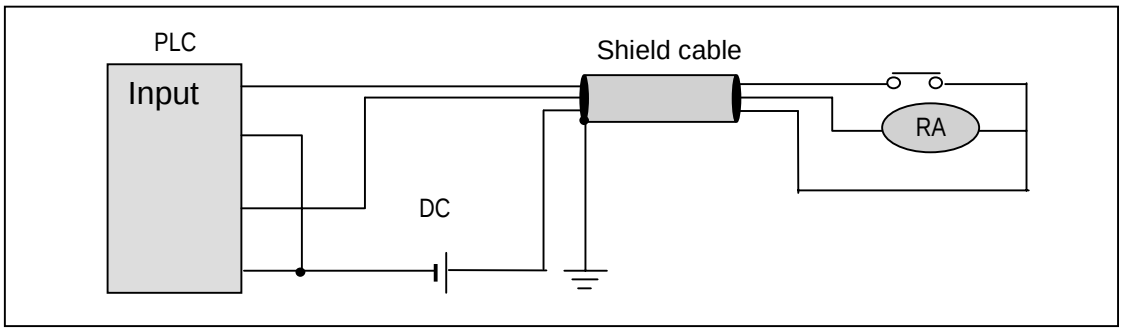
9.2.2 Input and Output Devices Wiring

1)Applicable size of wire to the terminal block connector is 0.18 to 2 . However, it is recommended to use wire of 0.3 for convenience.

2) Separate the input and output lines.

3) I/O signal wires must be at least 100 (3.94 in) away from high voltage and large current circuit wires.

4) When the I/O signal wires cannot be separated from the main circuit wires and power wires, ground on the PLC side
with batch-shielded cables. Under some conditions it may be preferable to ground on the other side.



5) If wiring has been done with of piping, ground the piping.

6) Separate the 24VDC I/O cables from the 110VAC and 220VAC cables.

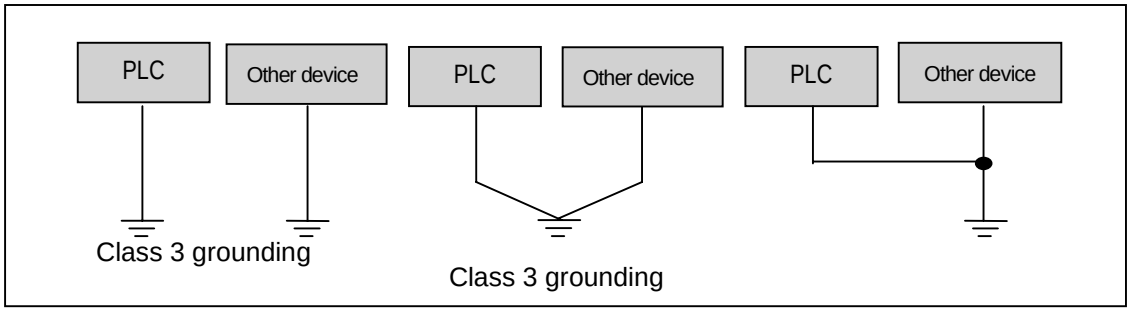
If wiring over 200□ (7.88 in) or longer distance, trouble can be caused by leakage currents due to line capacity. Refer to the section 11.4 Example.

9.2.3 Grounding

1)This PLC has sufficient protection against noise, so it can be used without grounding
except for special much noise. However, when grounding it should be done conforming to below items.

2)Ground the PLC as independently as possible. Class 3 grounding should be used
(grounding resistance 80Ωor less).

3)When independent grounding is impossible, use the joint grounding method as shown in the figure below (B).



(A)Independent grounding : Best (B) Joint grounding : Good (C) Joint grounding : Not allowed

4)Use 2□(14AWG) or thicker grounding wire. Grounding point should be as near as possible to the PLC to minimize the distance of grounding cable.

9.2.4 Cable Specifications for wiring

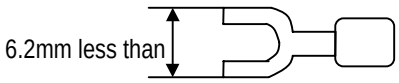
The specifications for wiring is as follows:

Kinds of external connection	Cable Specifications ( )	
	Minimum	Maximum
Digital Input	0.18 (AWG24)	1.5 (AWG16)
Digital Output	0.18 (AWG24)	2.0 (AWG14)
Analog Input / Output	0.18 (AWG24)	1.5 (AWG16)
Communication	0.18 (AWG24)	1.5 (AWG16)
Main power	1.5 (AWG16)	2.5 (AWG12)
Grounding	1.5 (AWG16)	2.5 (AWG12)

Be sure to use solidness terminal for power supply and I/O wirings.

- Be sure to use M3 type as terminal screw.
- Make sure that terminal screw is connected by 6□9□ · □ torque..
- Be sure to use fork shaped terminal screw as shown below.

cable solidness terminal (fork shaped)



Chapter 10 Maintenance

Be sure to perform daily and periodic maintenance and inspection in order to maintain the PLC in the best conditions.

10.1 Maintenance and Inspection

The I/O module mainly consist of semiconductor devices and its service life is semi-permanent. However, periodic inspection is requested for ambient environment may cause damage to the devices. When inspecting one or two times per six months, check the following items.

Check Items		Judgment	Corrective Actions
Ambient environment	Temperature	0 ~ + 55°C	Adjust the operating temperature and humidity with the defined range.
	Humidity	5 ~ 95%RH	
	Vibration	No vibration	Use vibration resisting rubber or the vibration prevention method.
Play of modules		No play allowed	Securely enrage the hook.
Connecting conditions of terminal screws		No loose allowed	Retighten terminal screws.
Change rate of input voltage		– 15% to 10%	Hold it with the allowable range.
Spare parts		Check the number of Spare parts and their Store conditions	Cover the shortage and improve the conditions

10.2 Daily Inspection

The following table shows the inspection and items which are to be checked daily.

Check Items		Check Points	Judgement	Corrective Actions
Connecting conditions of terminal block or extension cable		check for loose mounting screws	Screws should not be loose	Retighten Screws
		Check the distance between solderless terminals	Proper clearance should be provided	Correct
Indica -ting LED	PWR LED	Check that the LED is ON	ON(OFF indicates an error)	See chapter 11
	Run LED	Check that the LED is ON during Run	ON (flickering indicates an error)	See chapter 11
	ERR LED	Check that the LED is OFF during Run	OFF(ON indicates an error)	See chapter 11
	Input LED	Check that the LEO turns ON and OFF	ON when input is ON, OFF when input is off	See chapter 11
	Output LED	Check that the LEO turns ON and OFF	ON when output is ON, OFF when output is off	See chapter 11

10.3 Periodic Inspection

Check the following items once or twice every six months, and perform the needed corrective actions.

Check Items		Checking Methods	Judgment	Corrective Actions
Ambient Environment	Ambient temperature	Measure with thermometer and hygrometer measure corrosive gas	0 ~ 55 °C	Adjust to general standard (Internal environmental standard of control section)
	Ambient Humidity		5 ~ 95%RH	
	Ambience		There should be no corrosive gases	
PLC Conditions	Looseness, Ingress	The module should be move the unit	The module should be mounted securely.	Retighten screws
	dust or foreign material	Visual check	No dust or foreign material	
Connecting conditions	Loose terminal screws	Re-tighten screws	Screws should not be loose	Retighten
	Distance between terminals	Visual check	Proper clearance	Correct
	Loose connectors	Visual check	Connectors should not be loose.	Retighten connector mounting screws
Line voltage check		Measure voltage between input terminals	*85 ~ 264V AC *20~28V DC	Change supply power
Battery		Battery time and battery capacity life indicated, Change the reduction	Check total power failure If battery capacity time and the specified source	Battery capacity reduction should not be indicated battery when specified service life is exceeded
Fuse		Visual check	No melting disconnection	If fuse melting disconnection, change the fuse periodically because a surge current can cause heat

Chapter 11 TROUBLESHOOTING

The following explains contents, diagnosis and corrective actions for various errors that can occur during system operation.

11.1 Basic Procedures of Troubleshooting

System reliability not only depends on reliable equipment but also on short downtimes in the event of faults. The short discovery and corrective action is needed for speedy operation of system. The following shows the basic instructions for troubleshooting.

1) Visual checks

Check the following points.

- Machine operating condition (in stop and operating status)
- Power On/Off
- Status of I/O devices
- Condition of wiring (I/O wires, extension and communications cables)
- Display states of various indicators (such as POWER LED, RUN LED, ERR. LED and I/O LED). After checking them, connect peripheral devices and check the operation status of the PLC and the program contents.

2) Trouble Check

Observe any change in the error conditions during the following.

- Switch to the STOP position, and then turn the power on and off.

3) Narrow down the possible causes of the trouble where the fault lies, i.e.:

- Inside or outside of the PLC?
- I/O module or another module?
- PLC program?

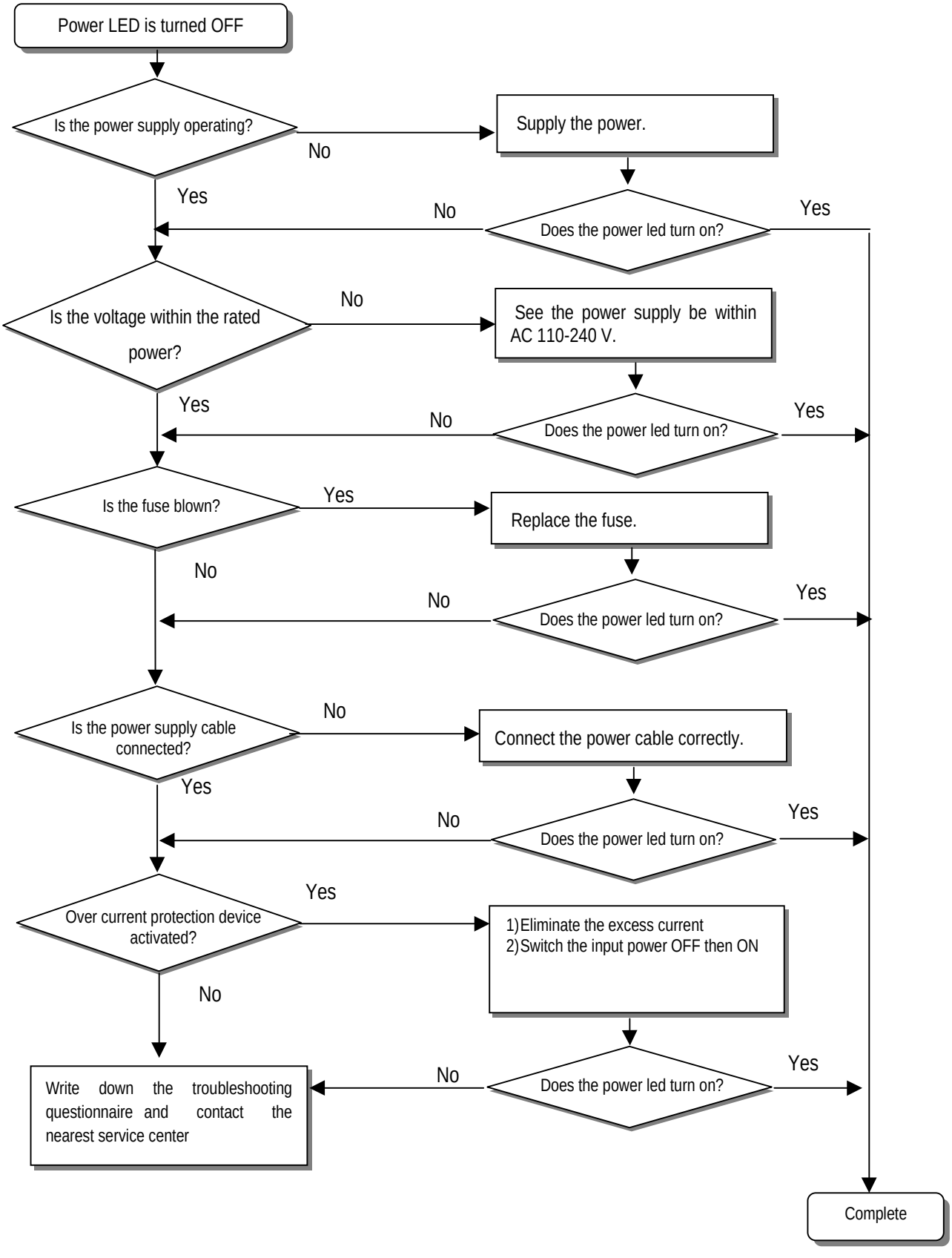
11.2 Troubleshooting

This section explains the procedure for determining the cause of troubles as well as the errors and corrective actions.

Is the power LED turned OFF?	⇨	Flowchart used when the POWER LED is turned OFF
Is the ERR LED flickering?	⇨	Flowchart used when the ERR LED is flickering
Are the RUN LED turned OFF?	⇨	Flowchart used when the RUN turned OFF.
I/O module doesn't operate properly	⇨	Flowchart used when the output load of the output module doesn't turn on.
Program cannot be written	⇨	Flowchart used when a program can't be written to the PLC

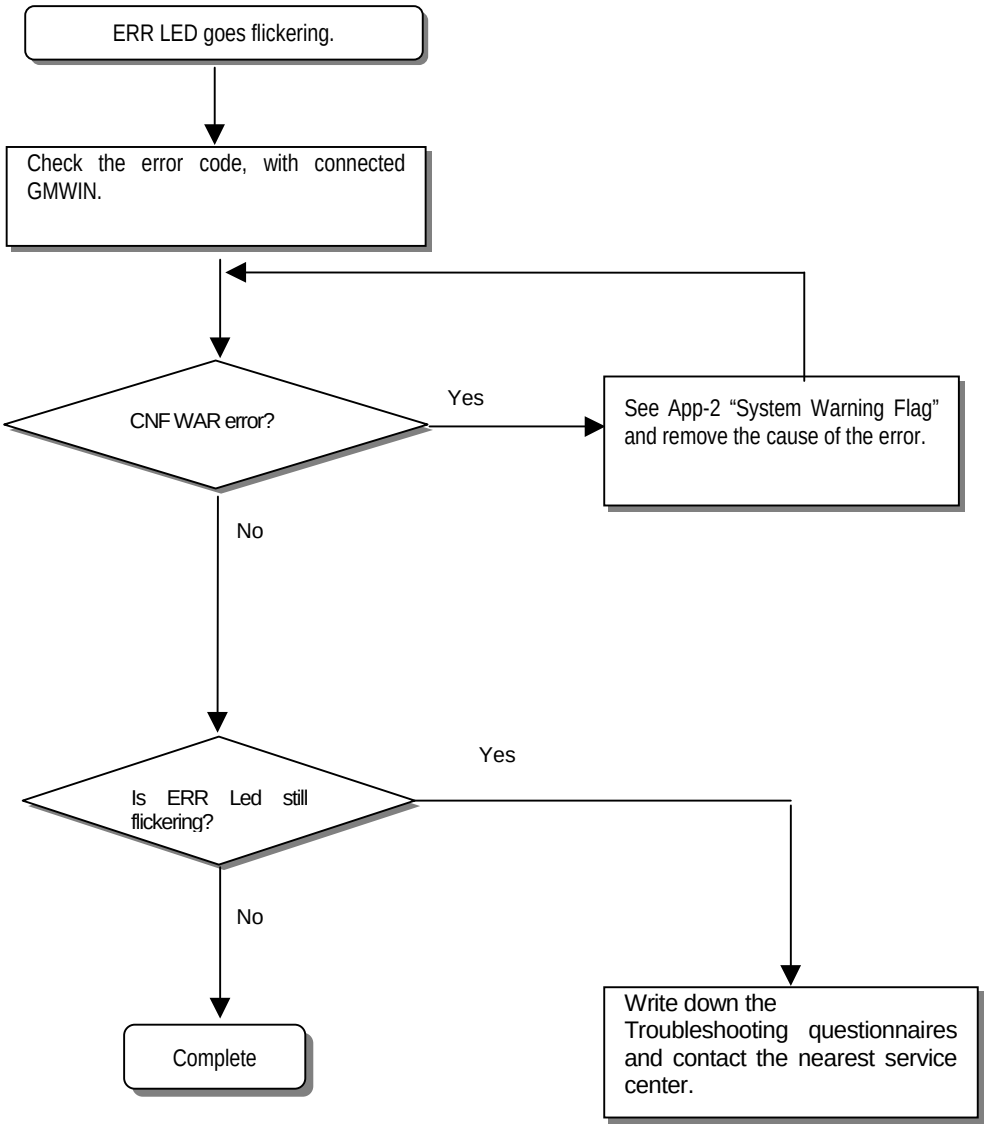
11.2.1 Troubleshooting flowchart used when the POWER LED turns OFF.

The following flowchart explains corrective action procedure used when the power is supplied or the power led turns off during operation.



11.2.2 Troubleshooting flowchart used when the ERR LED is flickering

The following flowchart explains corrective action procedure use when the power is supplied starts or the ERR LED is flickering during operation.

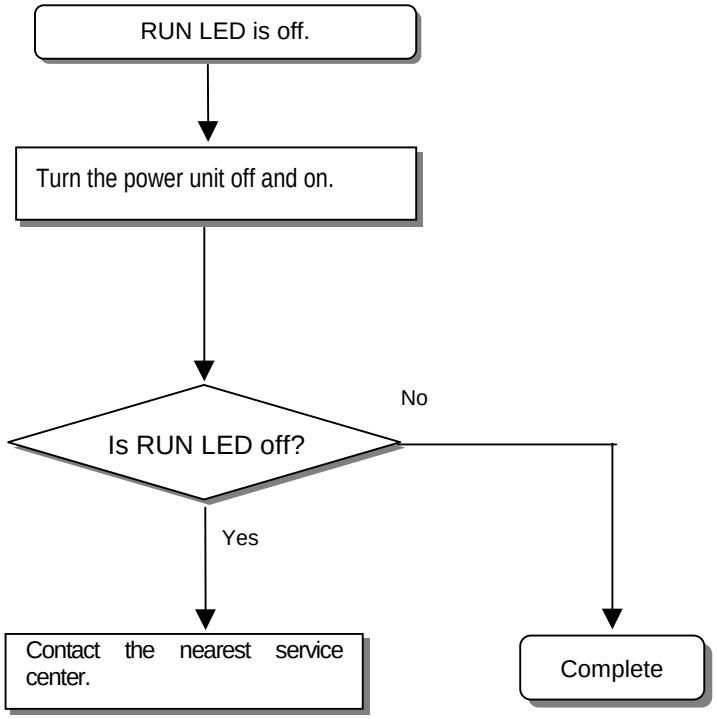

ERR LED goes flickering.
Check the error code, with connected GMWIN.
CNF WAR error?
Yes
See App-2 "System Warning Flag" and remove the cause of the error.
No
Is ERR Led still flickering?
Yes
No
Complete
Write down the Troubleshooting questionnaires and contact the nearest service center.

REMARK

Though CNF WAR appears, PLC system doesn't stop but corrective action is needed promptly. If not, it may cause the system failure.

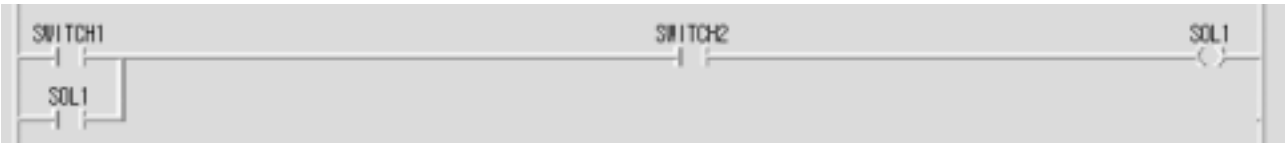
11.2.3 Troubleshooting flowchart used when the RUN turns off.

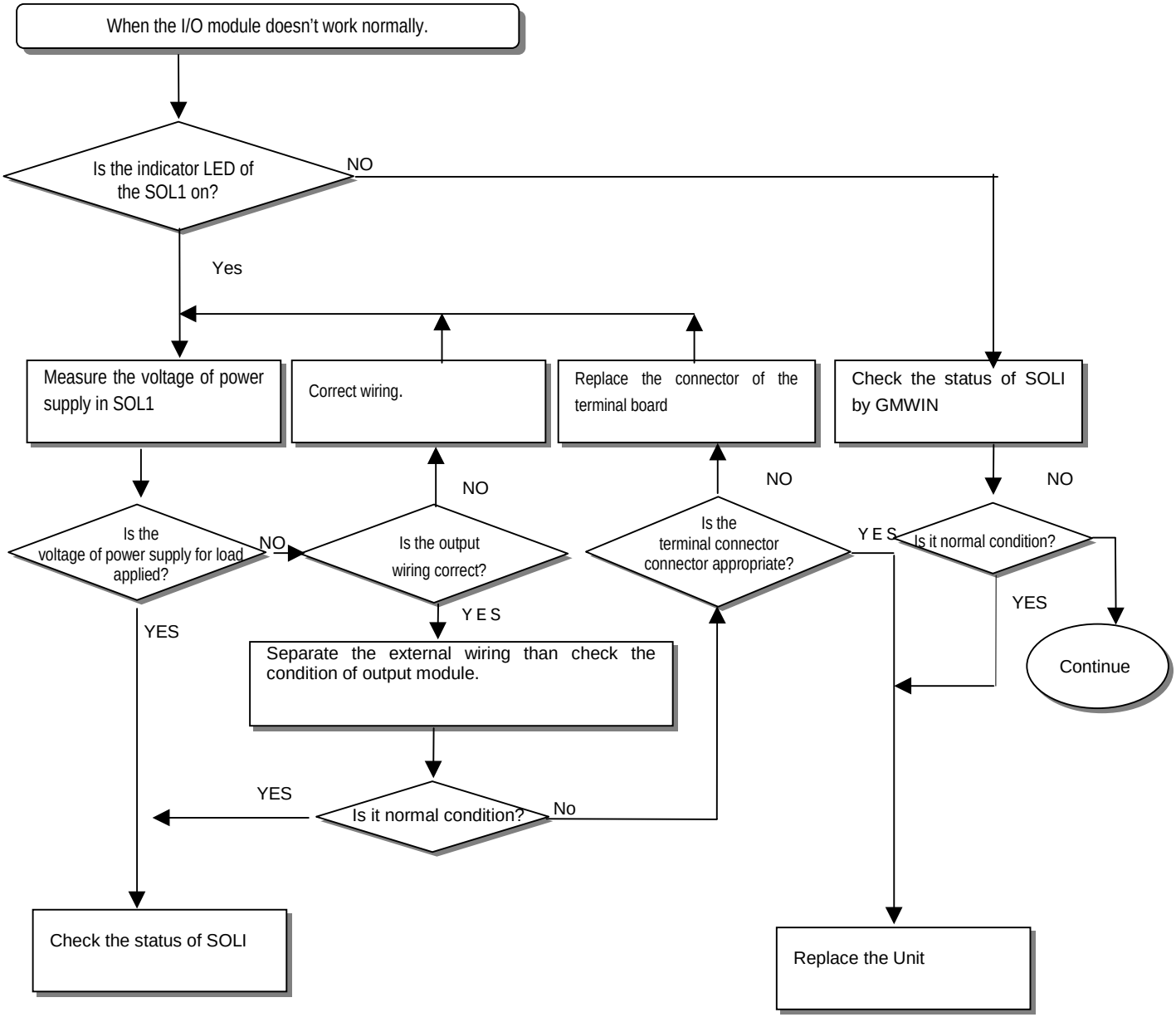
The following flowchart explains corrective action procedure to treat the lights-out of RUN LED when the power is supplied, operation starts or operation is in the process.

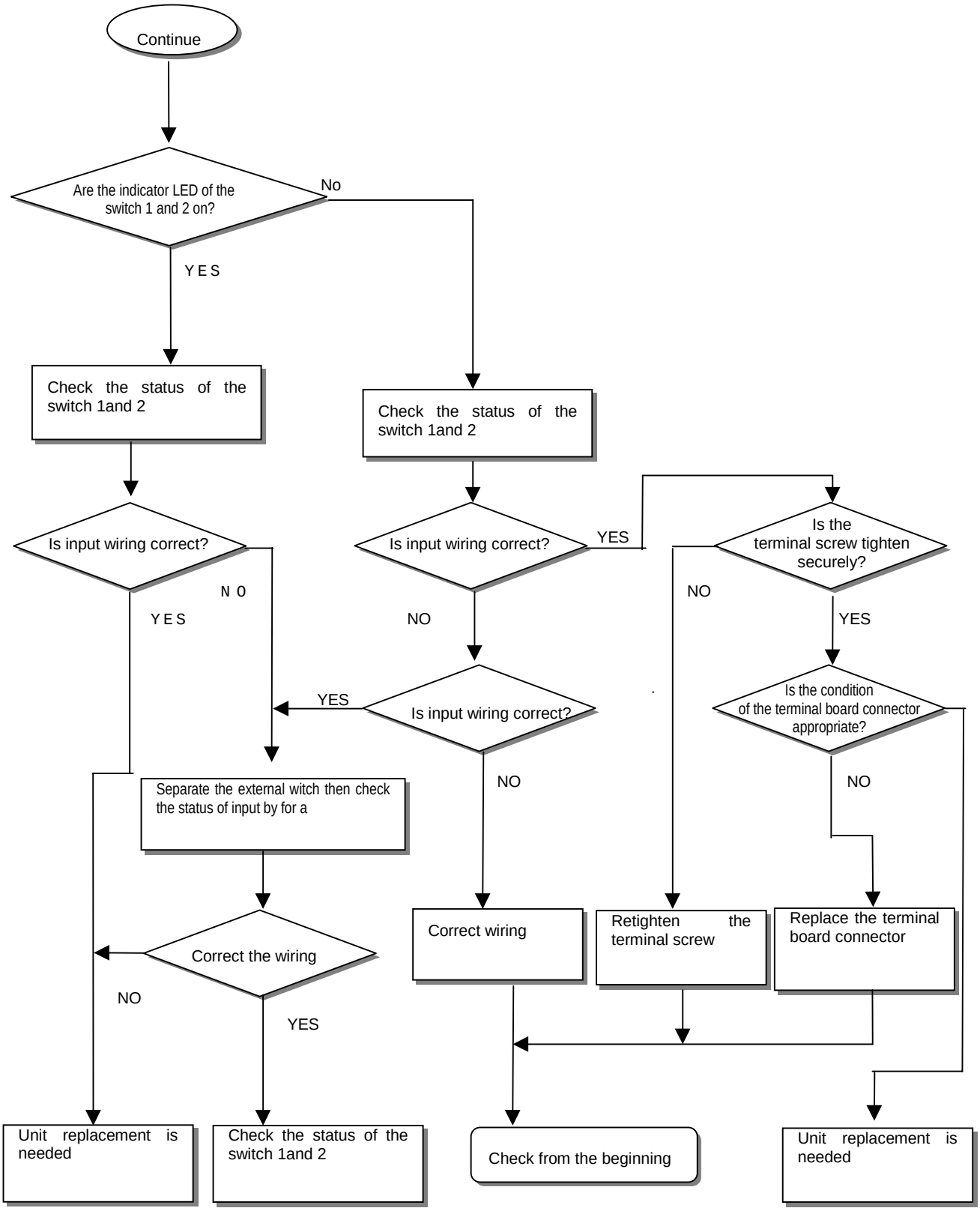


11.2.4 Troubleshooting flowchart used when the I/O part doesn't operate normally.

The following flowchart explains corrective action procedure used when the I/O module doesn't operate normally.

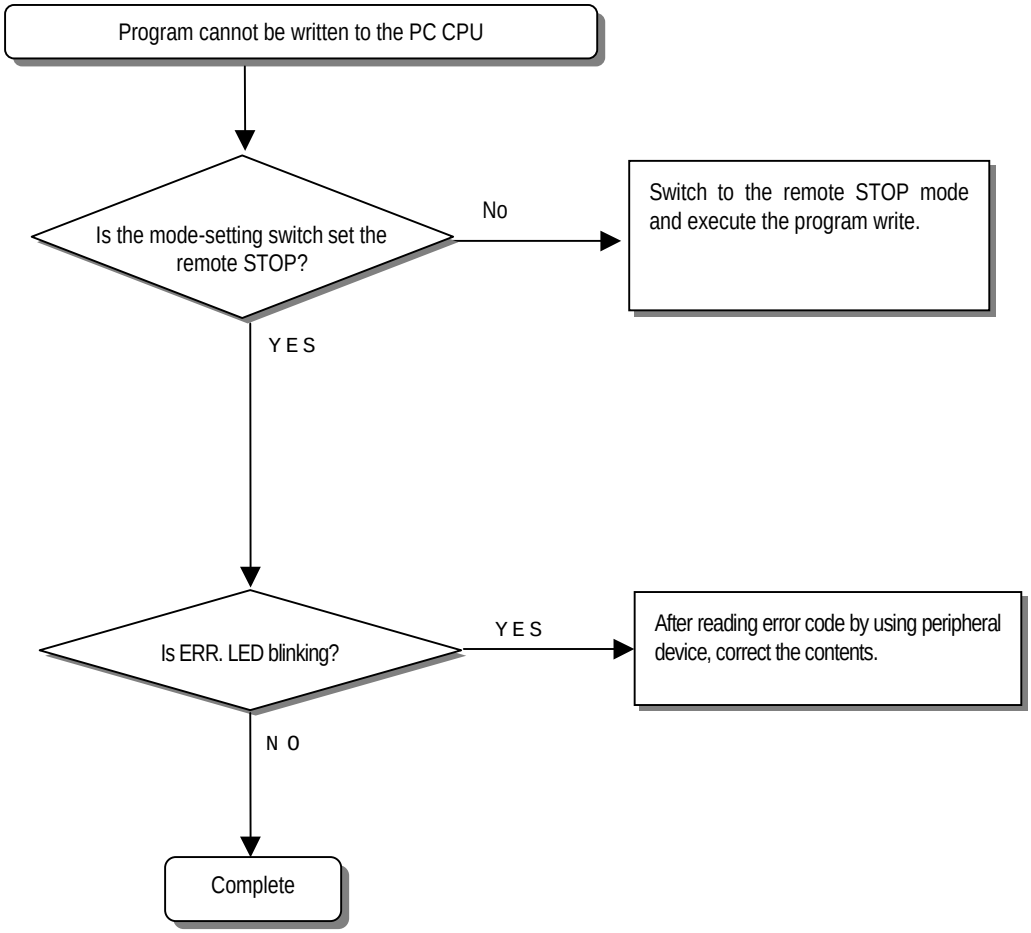





Continue
Are the indicator LED of the switch 1 and 2 on?
No
Y E S
Check the status of the switch 1and 2
Check the status of the switch 1and 2
Is input wiring correct?
N 0
Y E S
Is input wiring correct?
YES
NO
Is the terminal screw tighten securely?
NO
YES
Is input wiring correct?
YES
Is the condition of the terminal board connector appropriate?
NO
NO
Separate the external witch then check the status of input by for a
Correct the wiring
NO
YES
Correct wiring
Retighten the terminal screw
Replace the terminal board connector
Unit replacement is needed
Check the status of the switch 1and 2
Check from the beginning
Unit replacement is needed

11.2.5 Troubleshooting flowchart used when a program cannot be written to the CPU part

The following flowchart shows the corrective action procedure used when a program cannot be written to the PLC module.


Program cannot be written to the PC CPU
Is the mode-setting switch set the remote STOP?
No
Switch to the remote STOP mode and execute the program write.
YES
Is ERR. LED blinking?
YES
After reading error code by using peripheral device, correct the contents.
NO
Complete

11.3 Troubleshooting Questionnaire

When problems have been met during operation of the GM7 series, please write down this Questionnaires and contact the service center via telephone or facsimile.

• For errors relating to special or communication modules, use the questionnaire included in the User's manual of the unit.

Telephone & FAX No
Tell)
FAX)
Using equipment model:
Details of using equipment
CPU model:
OS version No.( ),
Serial No.( )
GMWIN version No. used to compile programs: ( )
General description of the device or system used as the control object:

5. The kind of the base unit:
– Operation by the mode setting switch ( ),
– Operation by the GMWIN or communications ( ),
– External memory module operation ( ),

6. Is the ERR. LED of the CPU module turned ON? Yes( ), No( )

7. GMWIN error message:

8. Used initialization program: initialization program ( )

9. History of corrective actions for the error message in the article 7:

10. Other tried corrective actions:

11. Characteristics of the error
• Repetitive(**):** Periodic( ), Related to a particular sequence( ), Related to environment( )
• Sometimes(**)**: General error interval:

12. Detailed Description of error contents:

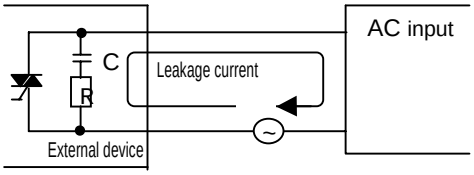
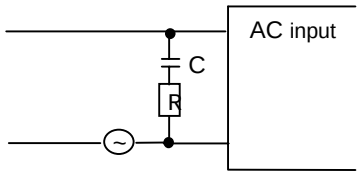
13. Configuration diagram for the applied system:

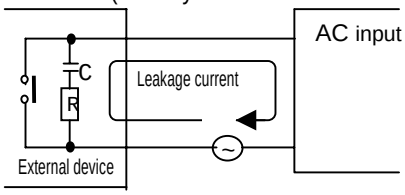
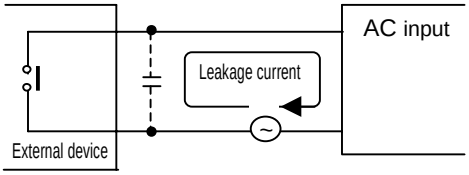
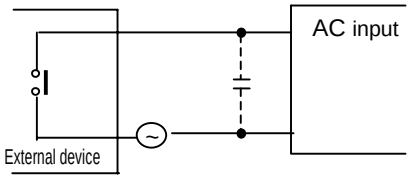
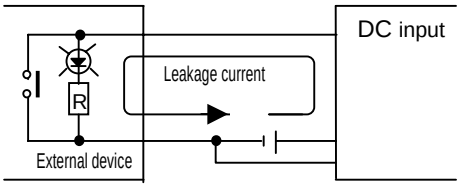
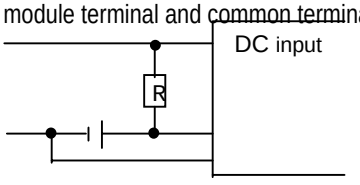
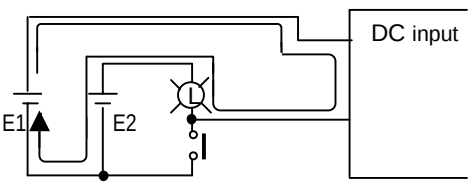
11.4 Troubleshooting Examples

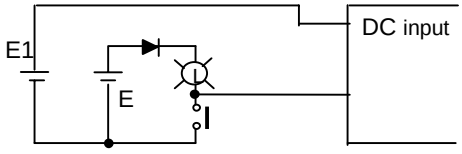
Possible troubles with various circuits and their corrective actions are explained.

11.4.1 Input circuit troubles and corrective actions

The followings describe possible troubles with input circuits, as well as corrective actions.

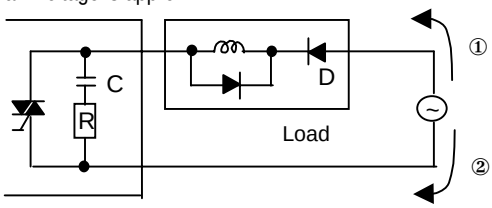
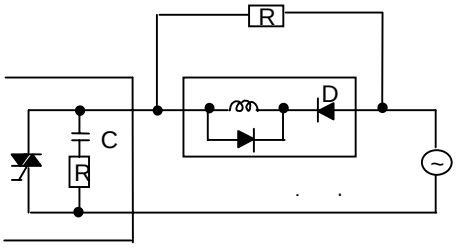
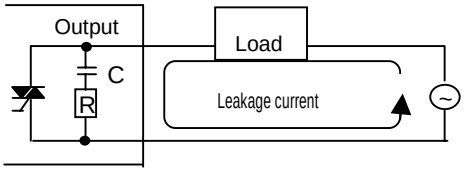
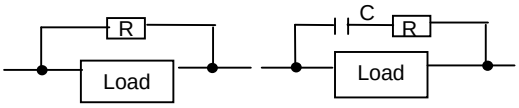
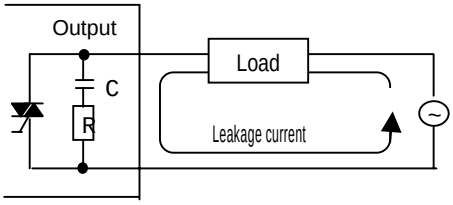
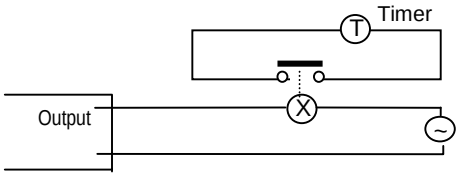
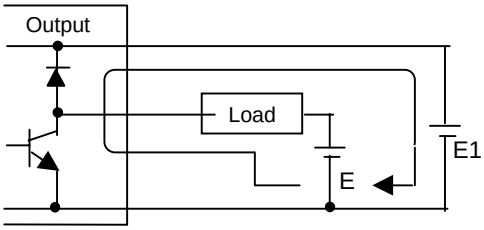
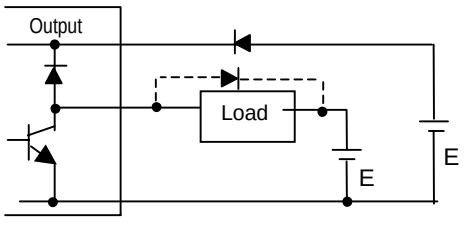
<table>
<tr><th>Condition</th><th>Cause</th><th>Corrective Actions</th></tr>
<tr><td>Input signal doesn't turn off.</td><td>Leakage current of external device
(Such as a drive by non-contact switch)
 
</td><td rowspan="2">• Connect an appropriate register and capacity, which will make the voltage lower across the terminals of the input module.
 

• CR values are determined by the leakage current value.
– Recommended value C : 0.1 ~ 0.47
R: 47 ~ 120 Ω (1/2W)
Or make up another independent display circuit.</td></tr>
<tr><td>Input signal doesn't turn off.
(Neon lamp may be still on)</td><td>Leakage current of external device
(Drive by a limit switch with neon lamp)
 
</td></tr>
<tr><td>Input signal doesn't turn off.</td><td>Leakage current due to line capacity of wiring cable.
 
</td><td>• Locate the power supply on the external device side as shown below.
 
</td></tr>
<tr><td>Input signal doesn't turn off.</td><td>Leakage current of external device
(Drive by switch with LED indicator)
 
</td><td>• Connect an appropriate register, which will make the voltage higher than the OFF voltage across the input module terminal and common terminal.
 
</td></tr>
<tr><td>Input signal doesn't turn off.</td><td>• Sneak current due to the use of two different power supplies.
 

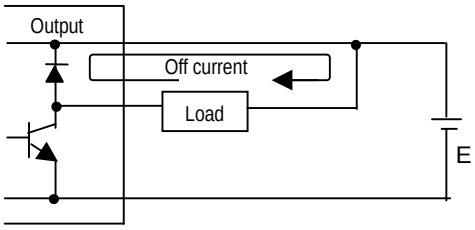
• E1 > E2, sneaked.</td><td>• Use only one power supply.
• Connect a sneak current prevention diode.
 
</td></tr>
</table>

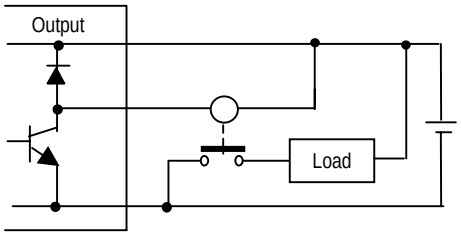
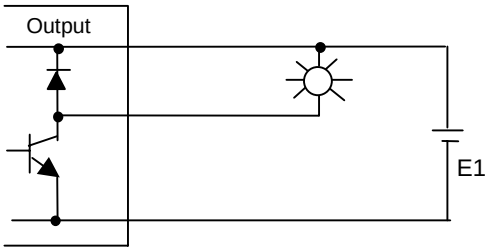
11.4.2 Output circuit troubles and corrective actions

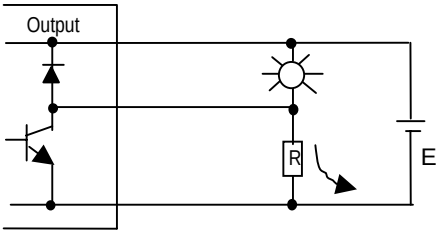
The following describes possible troubles with input circuits, as well as their corrective actions.

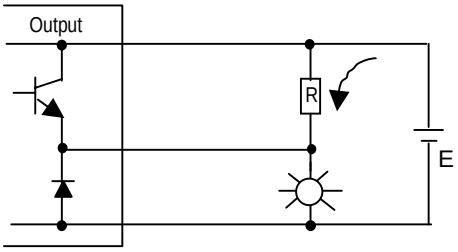
Condition	Cause	Corrective Action
When the output is off, excessive voltage is applied to the load.	•Load is half-wave rectified inside (in some cases, it is true of a solenoid) •When the polarity of the power supply is as shown in , C is charged. When the polarity is as shown in α, the voltage charged in C plus the line voltage are applied across D. Max. voltage is approx. 2□2.  *) If a resistor is used in this way, it does not pose a problem to the output element. But it may make the performance of the diode (D), which is built in the load, drop to cause problems.	• Connect registers of tens to hundreds K& across the load in parallel. 
The load doesn't turn off.	• Leakage current by surge absorbing circuit, which is connected to output element in parallel. 	• Connect C and R across the load, which are of registers of tens K&. When the wiring distance from the output module to the load is long, there may be a leakage current due to the line capacity. 
When the load is C-R type timer, time constant fluctuates.	• Leakage current by surge absorbing circuit, which is connected to output element in parallel. 	• Drive the relay using a contact and drive the C-R type timer using the since contact. • Use other timer than the C–R contact some timers have half-ware rectified internal circuits therefore, be cautious. 
The load does not turn off.	• Sneak current due to the use of two different power supplies.  E1<E2, sneaks. E1 is off (E2 is on), sneaks.	• Use only one power supply. • Connect a sneak current prevention diode.  If the load is the relay, etc, connect a counter-electromotive voltage absorbing code as shown by the dot line.

Output circuit troubles and corrective actions (continued).

<table>
<tr><th>Condition</th><th>Cause</th><th>Corrective actions</th></tr>
<tr><td>The load off response time is long.</td><td>• Over current at off state [The large solenoid current fluidic load (L/R is large) such as is directly driven with the transistor output.
 

• The off response time can be delayed by one or more second as some loads make the current flow across the diode at the off time of the transistor output.</td><td>• Insert a small L/R magnetic contact and drive the load using the same contact.
 
</td></tr>
<tr><td>Output transistor is destroyed.</td><td>Surge current of the white lamp
 

A surge current of 10 times or more when turned on.</td><td>• To suppress the surge current make the dark current of 1/3 to 1/5 rated current flow.
 

Sink type transistor output
 

Source type transistor output</td></tr>
</table>

11.5 Error code list

Error	Cause	Corrective action	Operation status	ERR. LED Flickering cycle	Diagnosis time	Restart mode
2	OS ROM error	Contact the A/S center if it reactively occurs when the power is re-applied.	Defect	0.4 sec.	When power is applied.	–
3	OS RAM error	Contact the A/S center if it reactively occurs when the power is re-applied.	Defect	0.4 sec.	When power is applied.	–
4	IC (RTC) error	Contact the A/S center if it reactively occurs when the power is re-applied.	Defect	0.4 sec.	When power is applied.	–
5	Fault processor	Contact the A/S center if it reactively occurs when the power is re-applied.	Defect	0.4 sec.	When power is applied.	–
6	Program memory fault	Contact the A/S center if it reactively occurs when the power is re-applied.	Defect	0.4 sec.	When power is applied.	–
7	Data memory fault	Contact the A/S center if it reactively occurs when the power is re-applied.	Defect	0.4 sec.	When power is applied.	–
10	Watch dog error due to RE-apply the power	Re-apply the power	Reset	–	During run	Cold
20	Program Memory backup error	Replace the battery if it has error check the program after cc-loading it, and if an error is detected replace the CPU module.	STOP	0.4 sec.	When power is applied.	Cold
22	Memory module program fault	Correct the memory module program and re-operate the system.	STOP	0.4 sec.	Change into the RUN mode	Cold
23	An normal program	Re-load the program and start it	STOP	0.4 sec.	Change into the RUN mode	Cold
30	Inconsistency between the specified modules by parameters and the loaded modules	Module type inconsistency error Refer to the flags (_IO_TYER, IO_TYER_N, IO_TYER [n]) and correct the in corrective slot, and restart the system.	STOP	0.4 sec.	Change into the RUN mode	Cold
31	Module dismounting or additional mounting during run	Module mounting/ dismounting error Refer to the flags (_IO_DEER, _IO_DEER_N, _IO_DEER [n]) and correct the in corrective slot, and restart the system.	STOP	0.4 sec.	When scan completes	Cold
32	Fuse disconnection during run	Fuse disconnection error Refer to the flags (_FUSE_ER, FUSE_ER_N, FUSE_ER [n]) and correct the in corrective slot, and restart the system.	STOP	0.4 sec.	When scan completes	Cold
33	Abnormal I/D module data access during run	I/O module read/write error Refer to the flags (_SP_IFER, _IP_IFER_N, _IP_IFER [n]) and restart the system.	STOP	0.4 sec.	When scan completes During execution of program	Cold

Error	Cause	Corrective action	Operation status	ERR. LED Flickering cycle	Diagnosis time	Restart mode
34	Abnormal special link module data access during run	Special/link module interface error Refer to the flags (_SP_IFER, _IP_IFER_N, _IP_IFER [n]) and restart the system.	STOP	0.4 sec.	When power is applied. When scan completes During execution of program	Cold
40	During run, Scan time over than the scan delay time specified by parameters	Check the scan delay time specified by parameters and correct the parameters or the program, and then restart the program.	STOP	0.4 sec.	During execution of program	Cold
41	Unreadable instructions in the user program.	Re-load the program and restart it.	STOP	0.4 sec.	During execution of program	Cold
50	External device fatal error.	Refer to the external device fatal error. Flag (ANNUN_ER, _ANC_ERR [n]) and correct the fault devices and then restart the system.	STOP	0.4 sec.	When scan completes	Cold
60	The 'E_STOP' function has been executed.	Correct the program so that the error elements that invoked the 'E_STOP' function can be eliminated in the program and restart the system (cold restart).	STOP	–	During execution of program	–
100	Communications module configuration error	If the number of computer 4 communications module is included, then adjust the maximum number with in 8.	STOP	0.4 sec.	When power is applied.	Cold
101	Special/ Communications module initialization failure	Adjust the number of high-speed communications modules loaded.	STOP	0.4 sec.	When power is applied.	Cold
500	Data memory backup error	If the batter has no error.	RUN	–	When power is applied. When scan completes	Cold
501	RTC data error	If the battery has no error, reset the time using the SMWIN.	RUN	2 sec.	When power is applied. When scan completes	–
502	Lower battery voltage	Replace the battery, which the power is being applied.	RUN	4 sec.	When power is applied. When scan completes	–

Appendix 1. System Definitions

1) Option

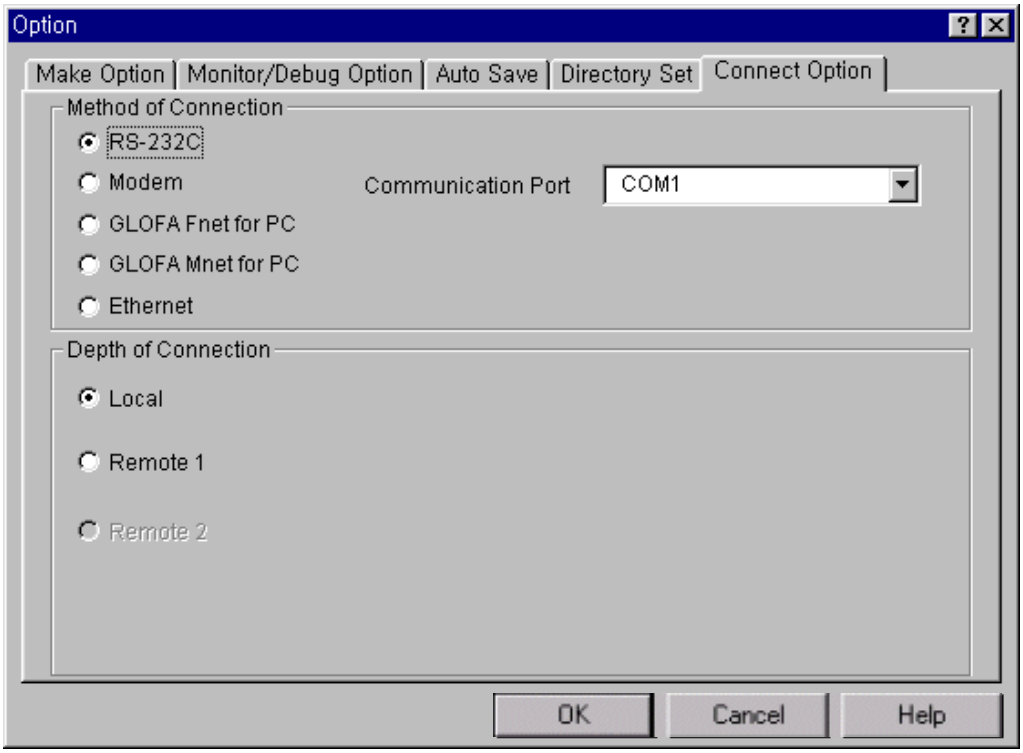
(1) Connect Option

You should set the communication port (COM1<4) to communicate with PLC.

∅ Select the ***Project-Option-Connect Option*** in menu.

∅ Default Connection is RS-232C interface.

For the detail information about **Connect Option**, refer to GMWIN Manual.



(2) Set the Directory of GMWIN

You can set directories for the files to be created in GMWIN.

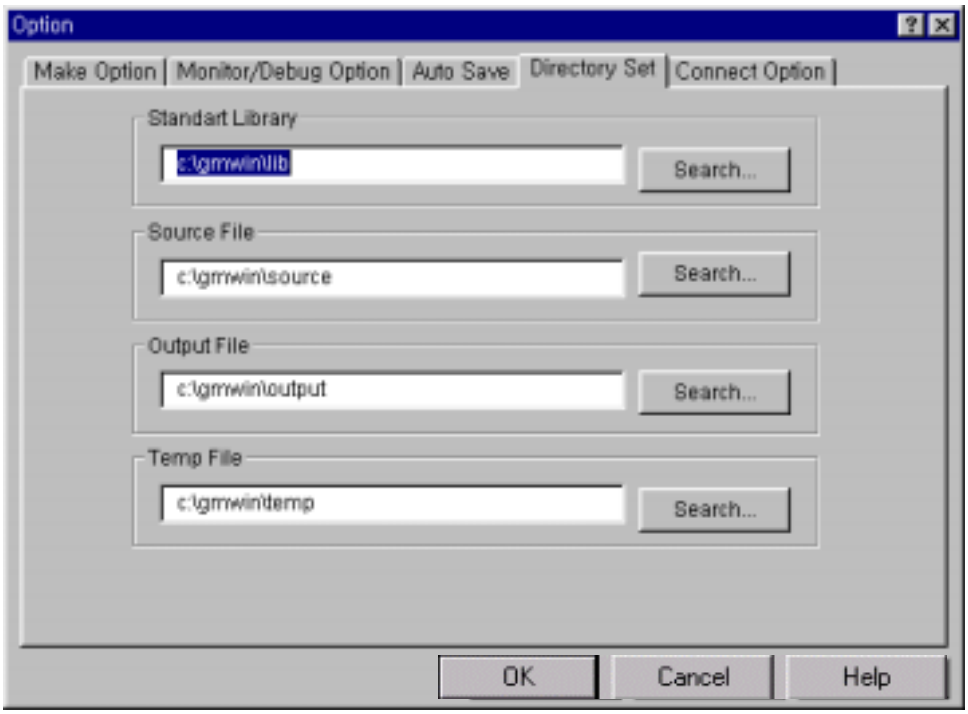
∅ Standard Library Directory : Libraries for GMWIN are located in this directory, and User Defined Libraries also do.

∅ Source File Directory : In Source File Directory, GMWIN saves source program files of program, function, function block and etc.

∅ Output File Directory : Object files are saved in this directory, which are created when source file is compiled.

∅ Temporary File Directory : GMWIN saves temporary file in this directory during the execution. For detailed descriptions refers to GMWIN manual.

For the detail information about **Directory Option**, refer to GMWIN Manual.



(3) Auto Save

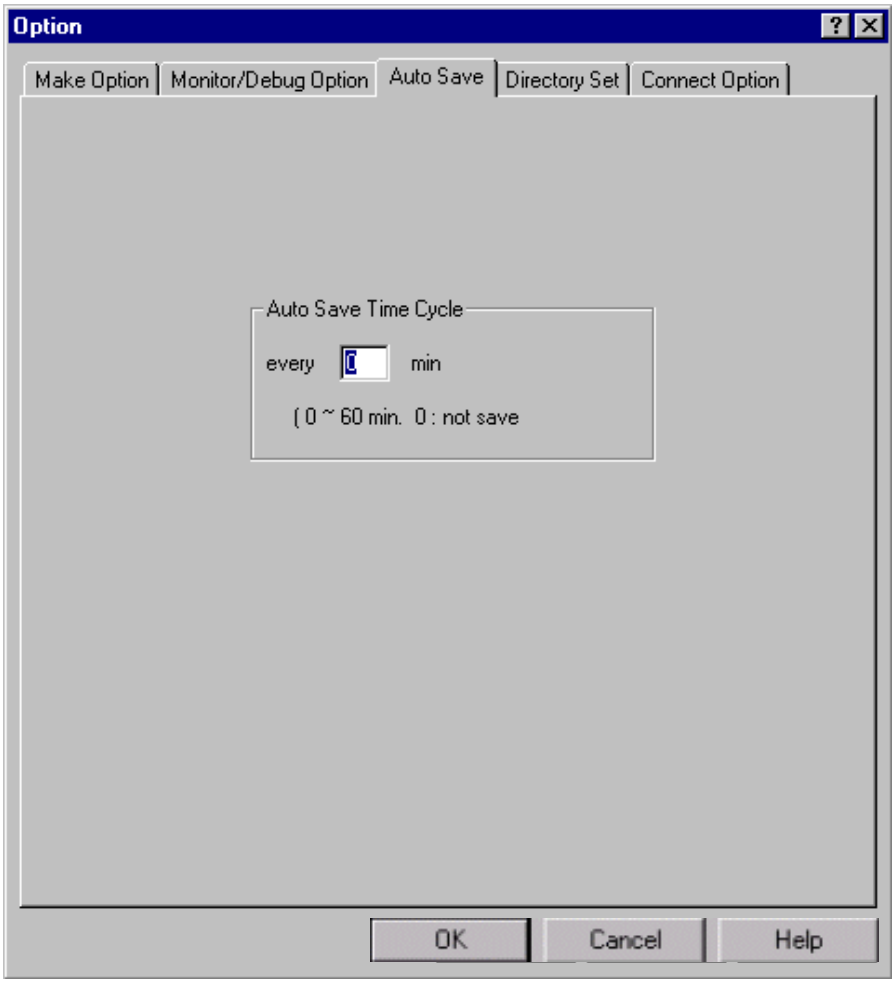
This function is to set the time interval for Auto saving.

Automatically saved file is saved with ".ASV" file in the current directory. The file is automatically deleted when the program window is closed. Therefore if a program cannot be saved by "Program Error" before program is not saved, you can recover some program by loading auto saved file.

∅ Select the ***Project-Option-Auto Save*** in menu.

∅ Enter the time interval for auto saving.(1 to 60 minutes is available, and if you enter '0', auto save function is disabled.)

∅ Click ***OK*** button.



(4) Monitor/Debug option

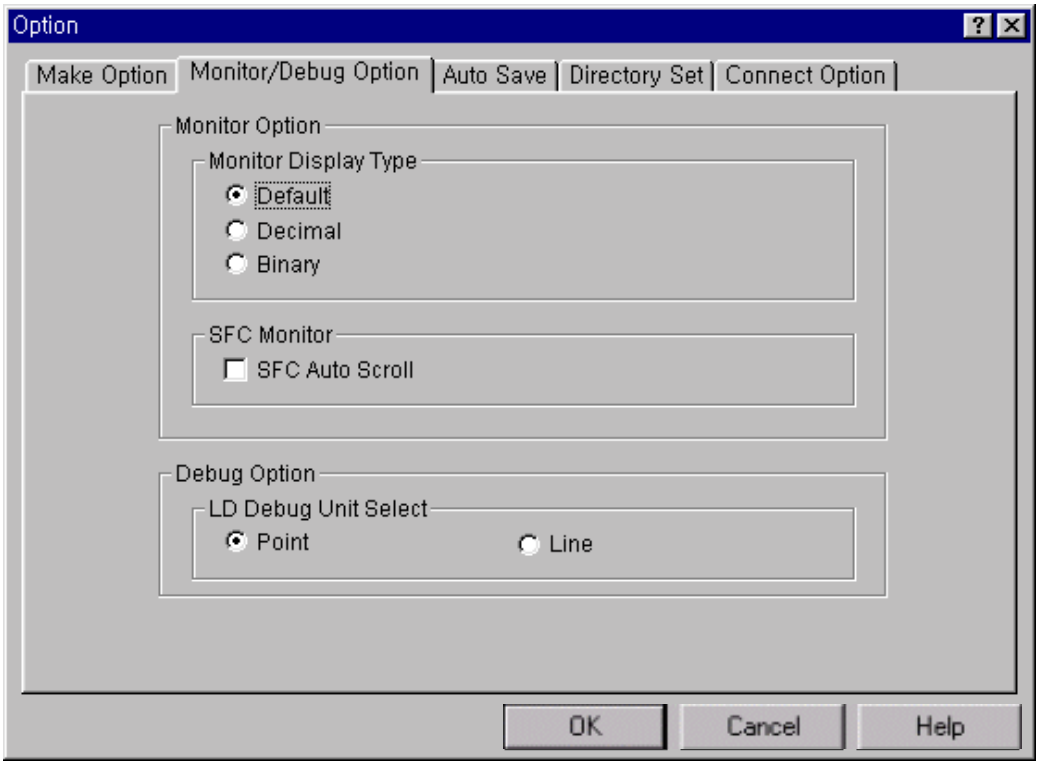
To set the whole options for monitoring,

∅ Select the ***Project-Option-Monitor/Debug Option*** in menu.

In monitoring mode, it allows you to change the display type of monitoring variable.

∅ Select the ***Project-Option-Monitor/Debug Option*** in menu.

∅ Select the desired display type in the ***Monitor Display Type***, click ***OK*** button.



Example) If you select the ***Default*** in ***Monitor Display Type***,
The value of variable monitoring is displayed as hexadecimal type like "16#*".

SFC Auto Scroll Option

∅ If you check the ***SFC Auto Scroll***, it scrolls automatically with monitoring position.

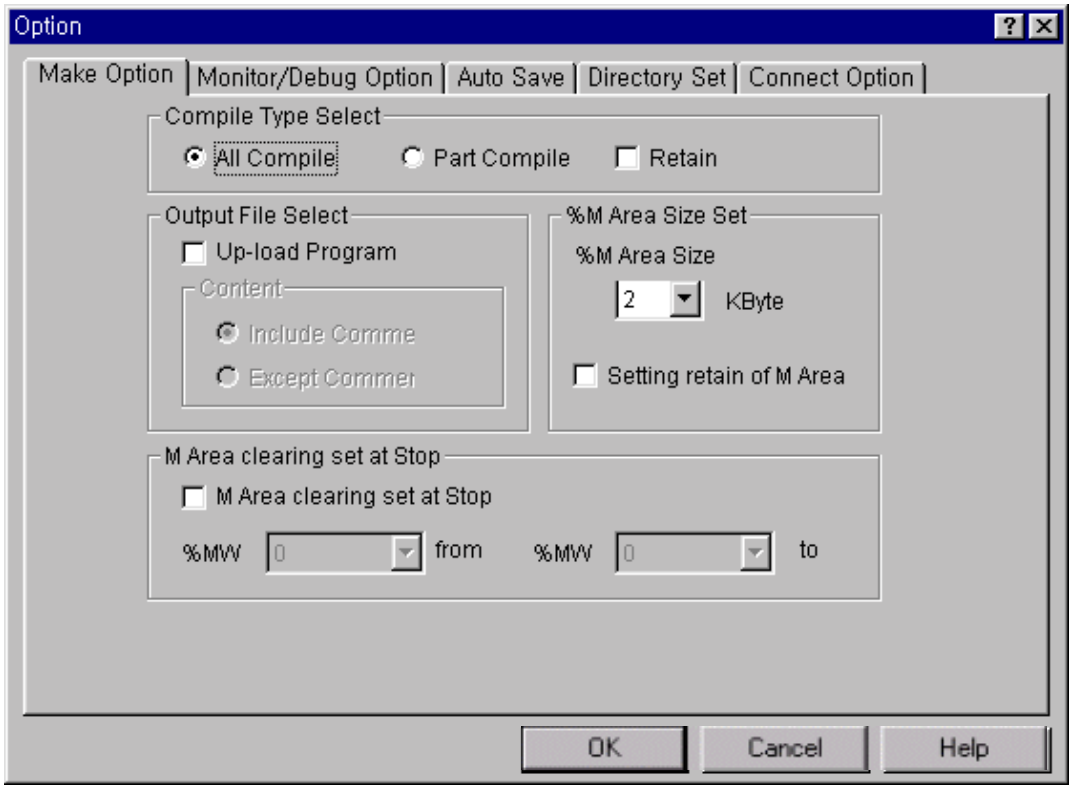
Debug Option

∅ When you debug LD, you can select the ***Point*** or ***Line*** in ***Debug Option*** menu.
If you select ***Point*** option, the debugging for the program is executed by one point.
If you select ***Line*** option, the debugging for the program is executed by one line.

(5) Make option

It allows you to select compile type, to select output file and to set the size of %M area in ***Make*** option.

∅ Select the ***Project-Option-Make Option*** in menu.



∅ ***Compile Type Select*** allows you to set the method of compile for the source program.

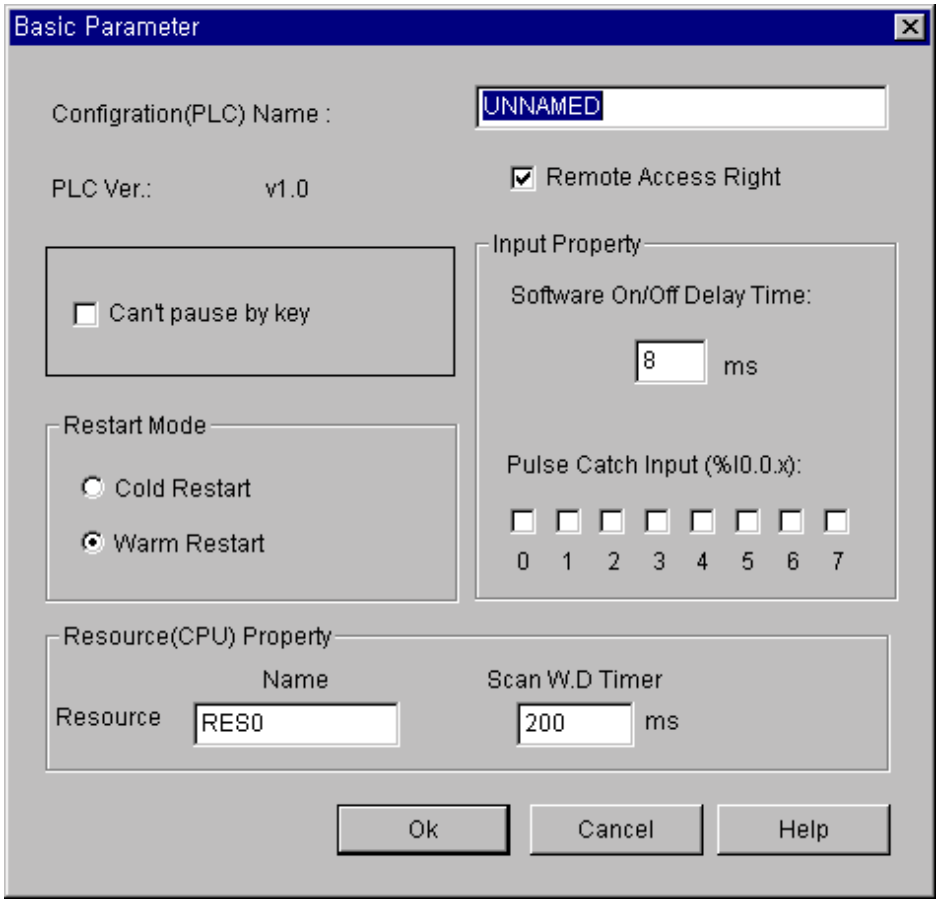
∅ If you select the ***Compile All*** in the ***Compile Type Select***, GMWIN compiles again from the first line regardless editing a source program.

∅ If you select the ***Part Compile*** in the ***Compile Type Select***, GMWIN compiles the updated part.

∅ You can select making variable table or upload program for the output file in ***Output File Select*** .

∅ You can select the size of %M in ***the %M Area Size Set***.

2) Basic Parameters



The basic parameters are necessary for operation of the PLC and used to allocate memory, set the restart mode and set the scan watch dog time, etc.

(1) Configuration (PLC) Name

- It is a representative name for the PLC system. It is used to designate this PLC system when a network system is configured using communication modules.

(2) Enabling/Disabling the control of the PLC via communications

- This parameter is used to enable or disable the remote control of this PLC system through the FAM or computer link module, etc. except for the GMWIN. If this parameter has been set to enable, change of the operation mode and download of programs are available via communications.

(3) Restart Mode

- This parameter is used to set the restart mode in the PLC system.

When the system re-starts, one of the 'cold restart' or 'warm restart' is selected in compliance with the parameter setting.

(4) Resource (CPU) Name

- Resource Name is the name that each CPU module configuring the PLC has. When configuring a network system the name is used to designate each CPU module that is used the system.
- Only one CPU module can be mounted in the GM7 series, therefore, only the resource 0 is valid.

(5) Scan Watch Dog Time

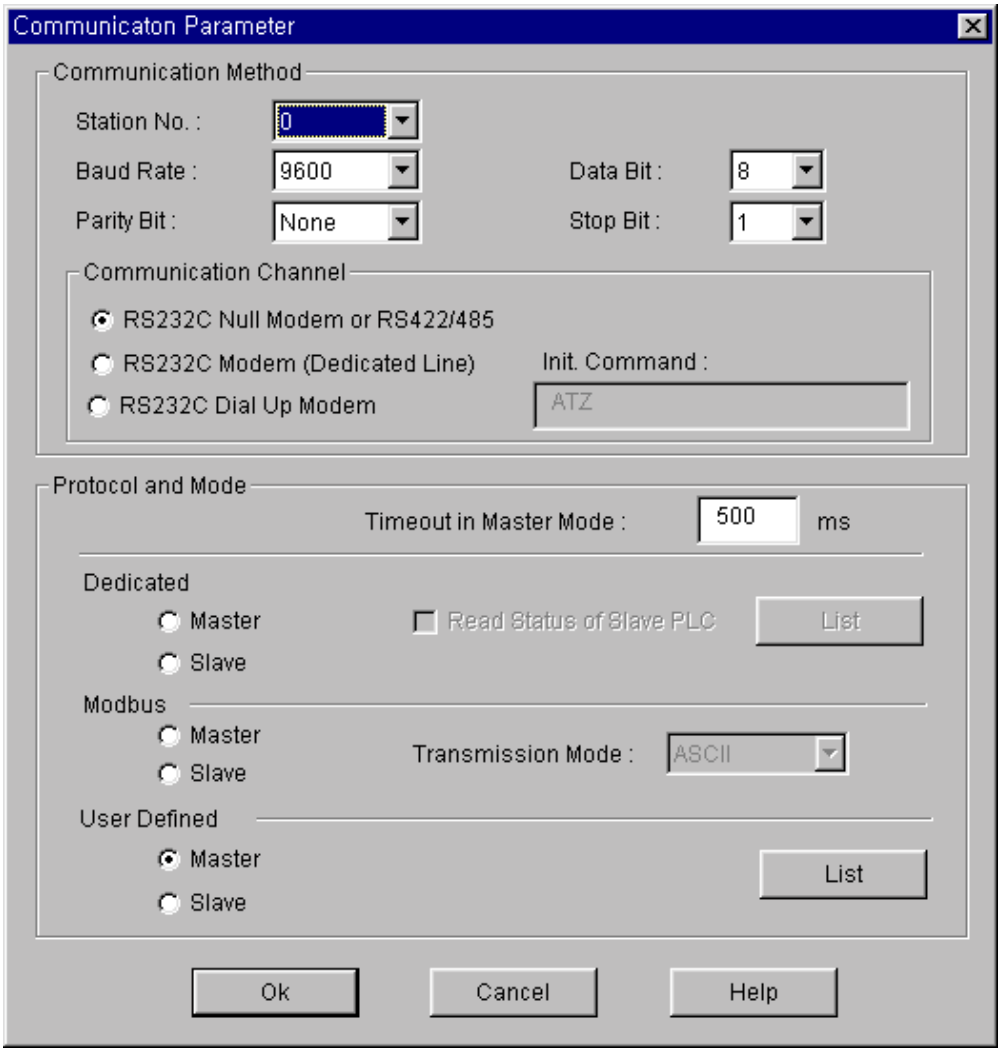
- This parameter is used to set the maximum allowable execution time of a user program in order to supervisor its normal or abnormal operation.
- Only one CPU module can be mounted in the GM7 series, therefore, scan watch dog is valid to only the resource 0.

(6) Input Property

- It's used to select contact point that will be used for setting input filter or as input pulse catch.

3) Communication parameter

This is a communication parameter to set regular sending/receiving stations, data and cycles to send and receive repeatedly. (For the detail information about Communication parameter, refer to 7.1.7 "Communication parameter setting")



(1) Station number(Station No.): 0 to 31

(2) Communication speed(Baud Rate): 1200,2400,4800,9600,19200,38400,57600bps.

(3) Data bit: 7 or 8 bits

(4) Parity bit: None, Even, odd

(5) Stop bit: 1 or 2 bit(s)

(6) Communication Channel

- RS232C null modem or RS422/485: Select this channel to communicate through GM7 base unit or Cnet I/F module (G7L-CUEC).
- RS232C modem(Dedicated Line): Select this channel to communicate through Cnet I/F module (G7L-CUEB).
- RS232C dial-up modem: Select this channel to communicate dial-up modem for modem communication, using Cnet I/F module (G7L-CUEB)

Remark

RS-232C modem(Dedicated Line) and RS232C dial up modem communication can be executed under RS-232C I/F module(G7L-CUEB)

(7) Master/slave: Select master to be major in the communications system.

(8) Time out

- The value of default is 500ms.
- Set the maximum cycle time for sending and receiving of the master PLC.
  - It may cause of communication error that lower setting value than maximum cycle time for sending and receiving.

(9) Reading slave PLC status.

- Select to read GM7 base unit status as slave designated. But do not choose this except for the monitoring of the slave status. It may cause to drop down the communication speed.

Appendix 2. Flag List

1) User Flag List

Keyword	Type	Write	Name	Description
_LER	BOOL	Enable	Operation error latch flag	Operation error latch flag by the program block(BP). Error indication occurred while executing a program block
_ERR	BOOL	Enable	Operation error latch flag	Operation error flag by the operation function (FN) or function block(FB). It is newly changed whenever an operation is executed.
_T20MS *	BOOL	–	20 ms Clock	These clock signals are used in the user programs, toggles on/off every half cycle. The clock signal can be delayed or distorted in accordance with program execution time as the signal toggles after scan has been finished, therefore, it is recommended that clock of enough longer than scan time be used. Clock signals starts from Off when the initialization program or scan program starts • Example : _T100MS clock
_T100MS *	BOOL	–	100 ms Clock	
_T200MS *	BOOL	–	200 ms Clock	
_T1S *	BOOL	–	1s Clock	
_T2S *	BOOL	–	2s Clock	
_T10S *	BOOL	–	10s Clock	
_T20S *	BOOL	–	20s clock	
_T60S *	BOOL	–	60s Clock	
_ON *	BOOL	–	Always On	Usable in user programs.
_OFF *	BOOL	–	Always Off	Usable in user programs.
_1ON *	BOOL	–	First scan On	Turn On only during the first scan after the operation has started.
_1OFF *	BOOL	–	First scan Off	Turn Off only during the first scan after the operation has started.
_STOG *	BOOL	–	Scan Toggle	Toggles On/Off at every scan while a user program is being executed. (On at the first scan)
_INT_DONE	BOOL	Enable	Initialization Program Complete	If this flag is set to on in the initialization program in an user program, the initialization program stop its operation and the scan program will starts.
_INT_DATE	DATE	–	RTC present date	Date Data of standard format (Reference date – Jan. 1, 1984)
_RTC_TOD	TOD	–	RTC present time	Time Data(Reference time – 00:00:00)
_RTC_WEEK	UNIT	–	RTC present day	Day data (0: Monday, 1:Thuesday, 2: Wednesday, 3: Thursday, 4: Friday, 5: Saturday, 6:Sunday)

Remark

1) Flags with the mark '*' are initialized when the initialization program starts, and after its execution has been competed the flags will change in accordance with the restart mode set.
2) RTC related flags could be used if only the optional module for RTC is installed.

2) Representative System Error Flag List

Keyword	Type	Bit No.	Name	Description
_CNF_ER	WORD	Represent --ative keyword	System error (fatal error)	This flag handles the following operation stop error flags in batch.
_IO _DEER	BOOL	Bit 2	Module loading/unloading error	This representative flag indicates that module configuration of each slot has been changed during operation. (Refer to _IO_DEER_N and _IO_DEER[n])
_IO _RWER	BOOL	Bit 4	I/O module read/write error	This representative flag indicates that a I/O module does normally executes read/write. (Refer to _IP_RWER_N and _IP_IFER[n])
_SP _IFER	BOOL	Bit 5	Special/communicat-ions module interface error	This representative flag indicates that special or communications module has failed in initialization or normal interface is impossible due to module malfunction. (Refer to _IP_IFER_N and _IP_IFER[n])
_ANNUN_ER	BOOL	Bit 6	External device fatal fault detection error	This representative flag indicates that an external device has fatal error. The error code has been written to _ANC_ERR[n].
–	–	Bit 7	–	–
_WD_ER	BOOL	Bit 8	Scan watch dog error	This flag indicates that the scan time of a program has overrun the scan watchdog time specified by the parameter.
_CODE_ER	BOOL	Bit 9	Program code error	This flag indicates that an unreadable instruction has been met while executing an user program.
_STACK_ER	BOOL	Bit 10	Stack overflow error	This flag indicates that the stack is used out of its capacity(Overflow)
_P_BCK_ER	BOOL	Bit 11	Program error	This flag indicates that program execution is impossible due to destroyed memory or program error.

3) Representative System Warning Flag List

Keyword	Type	Bit No.	Name	Description
_CNF _WAR	WORD	Represent -ative keyword	System warning	This flag treats the below warning flags relating to continuous operation in batch.
_RTC_ERR	BOOL	Bit 0	RTC data error	This flag indicates that RTC DATA error.
_D_BCK_ER	BOOL	Bit 1	Data backup error	This flag indicates
_AB_SD_ER	BOOL	Bit 3	Abnormal shutdown	This flag indicates that the program had been stopped during restore from power failure due to causes such as power off, and then cold restart has been executed and the continuous operation which retains the data is impossible. Usable in the initialization program. Automatically reset when the initialization program has finished. (The same things given above will be applied when the program has been stopped by the 'ESTOP' function)
_TASK_ERR	BOOL	Bit 4	Task collision (plus cycle and external tasks)	This flag indicates that task collision has occurred as execution request for a same task had been repeatedly invoked. (Refer to the flag _TC_BMAP[n] and _TC_CNT[n])
_BAT_ERR	BOOL	Bit 5	Battery fault	This flag detects and indicates that the voltage of the battery, which is used to backup user programs and data memory, is lower than the defined value.
_ANNUN_WR	BOOL	Bit 6	External device warning detection	This representative flag indicates that the user program has detected an ordinary fault of external devices and has written it to the flag _ANC_WB [n].
–	–	Bit 7	–	–
_HSPMT1_E R	BOOL	Bit 8	Communication Parameter 1 error	This representative flag detects error of each Communication parameter when the Communication has been enabled and indicates that Communication cannot be executed. It will be reset when the Communication is disabled.

4) Detailed System Error and Warning Flag List

Keyword	Type	Data setting range	Name	Description
_IO_RWER_N	UINT	0 to 15	The number of slot where I/O module read/write occurred.	This flag detects that input modules of a slot cannot be normally read from or written to, and indicates the lowest slot No. of the detected slot numbers.
_ANC_ERR[n]	UINT	n : 0 to 7	External device fatal error	This flag detects fatal error of external devices and its content is written to this flag. A number that identifies error type will be written to each of the sixteen locations. (The number 0 is not allowed)
_ANC_WAR[n]	UINT	n : 0 to 7	External device ordinary error	If the user program indicates a warning on the flag _ANC_WB[n], the bit locations are sequentially written to _ANC_WAR[n] from _ANC_WAR[0] complying with their occurrence sequence.
_ANC_WB[n]	BIT	n: 0 to 127	External device ordinary error bit map	The user program detects ordinary error of external device and the errors are indicated on a bit map. (The number 0 is not allowed)
_TC_BMAP[n]	BIT	n : 0 to 7	Task collision bit map	The flag detects that task collision has occurred because, while a task was being executed or ready for execution, an execution request has occurred for the same task, indicates the errors on a bit map.
_TC_CNT[n]	UINT	n : 0 to 7	Task collision counter	This flag detects task collision occurrence time for each task when executing a user program, indicates the task collision occurrence time.
_BAT_ER_TM*	DATE & TIME	—	Batter voltage drop time	The first detection date and time of battery voltage drop are written to this flag. It will be reset if the battery voltage has been restored.
_AC_F_CNT	UINT	0 to 65535	Momentary power failure occurrence count	The accumulated momentary power failure occurrence times during operation in the RUN mode is written to this flag.
_AC_F_TM[n]*	DATE & TIME	n : 0 to 15	Momentary power failure history	The times of the latest sixteen momentary power failures are written.
_ERR_HIS[n]*		n : 0 to 15	Error history	The times and error codes of the latest sixteen errors are written to this flag. • Stop time : DATE & TIME (8 bytes) • Error code : UINT (2 bytes)
_MODE_HIS[n]*		n : 0 to 15	Operation mode change history	The times, operation modes and restart modes of the latest sixteen operation mode changes are written to this flag • Change time : DATE & TIME (8 bytes) • Operation mode : UINT (2 bytes) • Restart : UINT (2 bytes)

* Marked flags are loaded while RTC option module is used..

5) System Operation status Information Flag List

Keyword	Type	Data setting range	Name	Description
_CPU_TYPE	Unit	0 to 16	System type	GM1 : 0, GM2 : 1, (GM3 : 2, GM4 : 3) (FSM : 5,6)
_VER_NUM	Unit	-	O/S version No.	System O/S version No.
_MEM_TYPE	Unit	1 to 5	Memory module type	Type of program memory module (0: Unloading state, type : 0 to 5)
_SYS_STATE	Word	Represent-ative keyword	PLC mode and operation status	System operation mode and operation state information
		Bit 0	Local control	Operation mode change is possible only by mode change switch or GMWIIN
		Bit 1	STOP	CPU module operation state
		Bit 2	RUN	
		Bit 3	PAUSE	
		Bit 4	DEBUG	
		Bit 5	Operation mode change factor	Operation mode change by mode change switch
		Bit 6	Operation mode change factor	Operation mode change by GMWIN
		Bit 7	Operation mode change factor	Operation mode change by remote GMWIN
		Bit 8	Operation mode change factor	Operation mode change by communications
		Bit 9	STOP by STOP function	Operation in the RUN mode is stopped by STOP function after the scan has finished
		Bit 10	Force input	Input junction force On/Off is being executed.
		Bit 11	Force output	Output junction force On/Off is being executed
		Bit 12	STOP by ESTOP function	Operation in the RUN mode is directly stopped by ESTOP function.
		Bit 13	-	-
		Bit 14	During monitoring	External monitoring is being executed for programs or variables
		Bit 15	Remote mode ON	Operation in the remote mode
_GMWIN_CNF	Byte	Representative keyword	GMWIN connection state	Connection state between CPU module and GMWIN
		Bit 0	Local GMWIN connection	Local GMWIN connection state
		Bit 1	Remote GMWIN connection	Remote GMWIN connection state
		Bit 2	Remote communications connection	Remote communications connection state

5) System Operation status Information Flag List (continued)

Keyword	Type	Data setting range	Name	Description
_RST_TY	Byte	Representative keyword	Restart mode information	Restart type of program which is being executed in present. (History)
		Bit 0	Cold restart	See the Section 4.5.1
		Bit 1	Warm restart	
		Bit 2	Hot restart	
_INIT_RUN	Bool	-	During initialization	An initialization program written by the user is being executed
_SCAN_MAX	Unit	-	Maximum scan time (ms)	Maximum scan time is written during operation.
_SCAN_MIN	Unit	-	Minimum scan time (ms)	Minimum scan time is written during operation.
_SCAN_CUR	Unit	-	Present scan time (ms)	Present scan time is continuously updated during operation.
_RTC_TIME[n]*	BCD	N : 0 to 7	Present time	BCD data of present time of RTC (Example : 96-01-12-00-00-00-XX) _RTC _TIME[0] : year, _RTC _TIME[1] : month, _RTC _TIME[2] : day, _RTC _TIME[3] : hour, _RTC _TIME[4] : minute, _RTC _TIME[5] : second, _RTC _TIME[6] : day of the week, _RTC _TIME[7] : unused Day of the week : 0 : Mon., 1: Tue., 2: Wed., 3:Thur., 4:Fri., 5: Sat., 6:Sun.
_SYS_ERR	Unit	Error code	Error type	See the Section 12.5 Error Code List

* Marked flags are loaded while RTC option module is used.

6) System Configuration status Information Flag

(1) User Program Status Information

Keyword	Type	Data setting range	Name	Description
_DOMAN_ST	BYTE	Representative keyword	System S/W configuration information	GM1 : 0, GM2 : 1, (GM3 : 2, GM4 : 3, GM% : 4) (FSM : 5,6), Twofold : 16
		Bit 0	Basic parameter error	Checks and indicates Basic parameter error
		-	-	-
		Bit 2	Program error	Checks and indicates Program error
		-	-	-
		Bit 4	Communication parameter error	Checks and indicates High speed link parameter error

(2) Operation Mode change switch Status Information

Keyword	Type	Data Setting range	Name	Description
_KEY_STATE	BYTE	Representati-ve keyword	Mode setting switch position	Indicates the state mode setting switch of CPU module
		Bit 0	KEY_STOP	Indicates that the mode setting switch is in the STOP state.
		Bit 1	KEY_RUN	Indicates that the mode setting switch is in the RUN state.
		Bit 2	KEY_PAUSE/REMOTE	Indicates that the mode setting switch is in the PAUSE/REMOTE state.

Appendix 3. Function / Function Block List

1) Function List

Name	Function	Size of PB (Byte) *1	Size of library (Byte) *2	Processing speed (μs) *3
				GM7
ABS (int)	Absolute value operation	36	-	2.0
ADD (int)	Addition	24	-	1.5
AND (word)	Logical multiplication	16	-	1.0
DIV (int)	Division	24	-	2.5
DIV (dint)	Division	24	-	3.3
EQ (int)	Equality' comparison	20	-	1.3
LIMIT (int)	To output upper and lower limits	24	848	4.8
MAX (int)	To output the maximum input value	24	1076	5.9
MOVE	To cop data	8	-	0.5
MUL (dint)	Multiplication	24	-	3.3
MUL (int)	Multiplication	24	-	2.5
ROL	To rotate left	20	136	3.7
BCD_TO_DINT	Conversion of BCD type into DINT	12	264	8.5
BCD_TO_INT	Conversion of BCD type into INT type	12	160	6.9
BCD_TO_SINT	Conversion of BCD type into SINT type	12	108	5.3
BYTE_TO_SINT	Conversion of BYTE type into SINT type	8	-	0.5
DATE_TO_STRING	Conversion of DATE type into string	32	314	20.8
DINT_TO_INT	Conversion of DINT pe into INT type	48	-	2.2
DINT_TO_BCD	Conversion of DINT type into BCD type	12	156	8.8
DT_TO_DATE	Conversion of DT type into DATE type	16	4	1.1
DT_TO_TOD	Conversion of DT type into TOD type	16	12	1.4
DT_TO_STRING	Conversion of DT type into string	36	620	21.0
DWORD_TO_WORD	Conversion of DWORD type into WORD	8	-	0.5
INT_TO_DINT	Conversion of INT type into DINT type	12	-	0.7
INT_TO_BCD	Conversion of INT type into BCD type	12	100	7.2
NUM_TO_STRING (int)	Conversion of number into string	24	580	15.9
SINT_TO_BCD	Conversion of SINT type into BCD type	12	76	5.9
STRING_TO_INT	Conversion of string info NT type	12	1264	28.9
CONCAT	To concatenate strings	48	172	5.9
DELETE	To delete string	40	172	6.9
EQ (str)	'Equality' comparison	32	948	8.3
FIND	To find a string	24	220	7.9
INSERT	To insert a string	48	160	8.9
LEFT	To obtain the left part of a string	36	100	6.4
LEN	To obtain the length of a string	12	40	4.5
LIMIT (str)	To output upper or lower limits	60	794	8.9
MAX (str)	To output the maximum input value	52	1076	8.4
MID	To obtain the middle part of a string	40	188	7.1
REPLACE	To replace a string with another	52	288	7.9
RIGHT	To obtain the hr part of a scan	36	164	6.9
ADD_TIME (time)	Time addition	20	148	5.6
DIV_TIME (i1=time)	Time division	20	152	6.9

Remark

1) The items marked with '∗ ' has following meaning.
 ∗ 1: The size of the program memory which a program occupies when it uses the function once
 ∗ 2: The size of the program memory which a program occupies only one time though it uses the function many times
 ∗ 3: of IL programs (2 input variables, 10 strings)
2) The above shows the function lists when programs are written with IL (instruction List) language.
 If programs are written with LD (Ladder diagram), the following differences occur.
 (1) 16 bytes will be added to the size of the PB.
 (2) In non-execution, 0.4 will be added to the processing speed. In execution, 0.8 sec will be added.

2) Function Block List

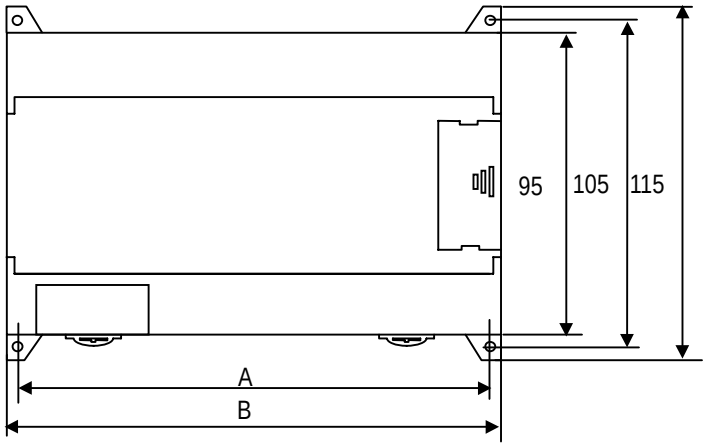
Name	Function	Size of PB (Byte)∗2	Size of library		Processing speed (μs) *4
			Size (Byte)∗3	Size of Instance memo∗ 3	GM7
CTU	Addition counter	24	92	6	3.8
CTUD	Addition/subtraction counter	32	168	6	4.4
F_TRIG	Descending edge detection	16	28	1	2.6
RS	Preference reset table	20	44	2	3.2
TON	ON delay timer	20	182	20	4.8

Remark

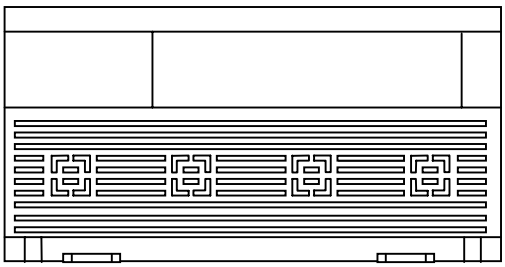
1) The items marked with '∗' has following meaning.
 ∗ 1: The size of the program memory which a program occupies when it uses the function once
 ∗ 2: The size of the program memory which a program occupies only one time though it uses the friction many times
 ∗ 3: The size of the program memory which a program occupies whenever it uses the function block once
2) The occupied memory size and processing speed of IL programs are same as LD programs.

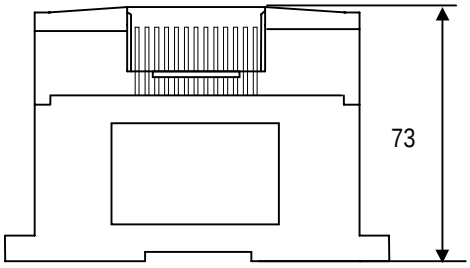
Appendix 4 External Dimensions (unit: mm)

1) Base unit



	A	B
10points	85	95
20 points	135	145
30points	135	145
40points	165	175
60points	215	225





2) Extension / Option modules